

CAN AHA MOMENTS BE FAKE? IDENTIFYING TRUE AND DECORATIVE THINKING STEPS IN CHAIN-OF-THOUGHT

Anonymous authors

Paper under double-blind review

ABSTRACT

Recent large language models (LLMs) can generate long Chain-of-Thought (CoT) at test time, enabling them to solve complex tasks. These reasoning traces are often assumed as a *faithful* reflection of the model’s internal thinking process, and used to monitor for unsafe intentions. However, we show that these reasoning traces can be decorative. We apply Average Treatment Effect (ATE) analysis to measure the step-wise causal influence of each reasoning step on the model’s final prediction. Based on this analysis, we design a *True Thinking Score (TTS)* to quantify how faithfully each step contributes to the answer. Our experiments show that LLMs often interleave between *true-thinking* steps (which are genuinely used to produce the final output) and *decorative-thinking* steps (which only give the appearance of reasoning but have minimal causal impact). Specifically, we reveal that only a small subset of the total reasoning steps have a high TTS that causally drive the model’s prediction: e.g., for the AIME dataset, only an average of 2.3% of reasoning steps in CoT have a $TTS \geq 0.7$ (range: 0–1) under the Qwen-2.5 model. Furthermore, we identify a TrueThinking direction in the latent space of LLMs. By steering along or against this direction, we can force the model to perform or disregard certain CoT steps when computing the final result. Finally, we highlight that self-verification steps in CoT (i.e., aha moments) can also be decorative, where LLMs do not truly verify their solution. Steering along the TrueThinking direction can force internal reasoning over these steps, resulting in a change in the final results. Overall, our work reveals that LLMs often verbalize reasoning steps without actually performing them internally, which undermines both the efficiency of LLM reasoning and the trustworthiness of CoT.

1 INTRODUCTION

Recent frontier LLMs can increasingly solve complex reasoning problems through test-time scaling, often by generating very long chains of thought (CoT) (Guo et al., 2025; Muennighoff et al., 2025; Snell et al., 2024; Jaech et al., 2024). In their long CoT, these models frequently exhibit an “aha moment”, where the model begins to *self-verify* its solution (e.g., “Wait, let’s re-evaluate ...”) (Guo et al., 2025). LLMs’ generated CoT is commonly assumed as a scratch pad where the model thinks out loud (Korbak et al., 2025). This has also motivated using CoT as a means to monitor LLMs and detect unsafe behaviors revealed in their CoT (Baker et al., 2025).

A central yet questionable assumption about CoT is its *faithfulness*: each verbalized step genuinely reflects the model’s *internal reasoning* and contributes to its final output. However, recent evidence shows this assumption is not always the case. Models may solve problems relying on hints (Chen et al., 2025b; Chua & Evans, 2025; Turpin et al., 2023) or their biases (Arcuschin et al., 2025) without verbalizing them in their CoT, and they may already know their final answers early before finishing generating the complete CoT (Ma et al., 2025; Wang et al., 2025; Yang et al., 2025). Such findings motivate the view that CoTs may act as *post-hoc rationalizations* (Arcuschin et al., 2025; Emmons et al., 2025), where LLMs first pre-determine their answers internally in their latent space and then generate reasoning steps to rationalize them. Such unfaithfulness of CoT raises concerns about relying on CoT for monitoring LLMs, as the verbalized reasoning may not reflect what a model truly

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107

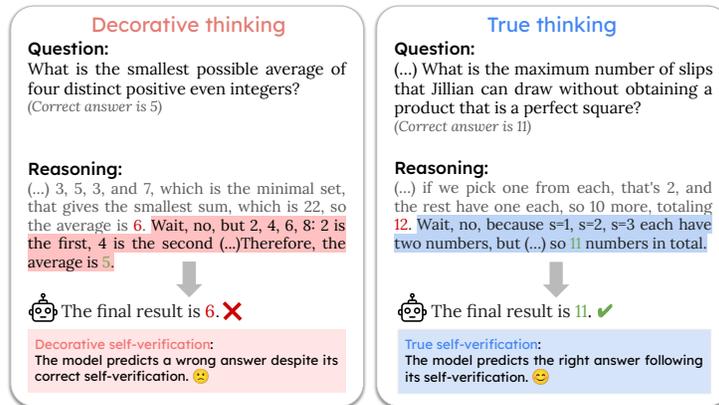


Figure 1: We find that reasoning steps in CoT may not always be *true thinking* but function as *decorative thinking* where the model internally is not using those steps to compute its answer. Taking self-verification steps as an example (known as “Aha moments” where LLMs rethink their solution with phrases like “wait”), we randomly perturb the numerical values in the reasoning steps preceding the “Aha moment”, and then re-prompt the model for the answer using the modified CoT. In the left example, although the model’s self-verification reasoning is correct, it ignores it and outputs the wrong answer after perturbation. In the right example, the model follows its self-verification and produces the correct result.

“thinks”. Although prior work has questioned the faithfulness of CoTs, a fine-grained, step-by-step analysis remains lacking. Therefore, in this study, we ask: *To what extent do LLMs truly think through each verbalized step in their CoT?*

To close this gap, we propose to measure the step-wise causality to probe whether an LLM is faithfully thinking as verbalized in its reasoning traces in CoT. We reveal that in a CoT, there are faithful **true-thinking** steps that causally affect the model’s prediction, and unfaithful **decorative-thinking** steps that the model does not actually perform internally and that make minimal causal contribution to its prediction (examples are shown in Figure 1). Crucially, a **true-thinking** step can causally contribute in two distinct ways as illustrated in Figure 2.

1. **Conjunctive (“and”)**: a step s and other steps before it (denoted as C) jointly determine the answer, as in many enumeration problems where all steps are important. Then, removing or corrupting s will flip the model’s initial prediction y^* . This is the regime primarily tested by prior work (Lanham et al., 2023; Tanneru et al., 2024; Wang et al., 2025; Yu et al., 2025), which infers faithfulness from the **necessity-in-context** effect of perturbing s alone.
2. **Disjunctive (“or”)**: either s or C already suffices to produce the correct answer. For example, s is a verification step or alternative solution for the results established in C . Here, perturbing s may leave model’s prediction unchanged because C still carries the solution. Prior works (Lanham et al., 2023; Tanneru et al., 2024; Wang et al., 2025; Yu et al., 2025) that only consider necessity may mislabel s in this case as “unfaithful” despite its genuine contribution.

To measure both roles, we extend Average Treatment Effect (ATE) (Rubin, 1974; Pearl, 2009) with two complementary interventions by conditioning on context C (steps before the step s): a **necessity** test $ATE_{nec}(1) = P(y^*|C, s) - P(y^*|C, s')$ that measures model’s confidence change before and after perturbing s under intact C , and a **sufficiency** test $ATE_{suf}(0) = P(y^*|C', s) - P(y^*|C', s')$ that perturbs s under corrupted C' . Averaging them yields our **True-Thinking Score (TTS)**, which considers steps that matter either jointly with context (the “and” case) or as an alternative route that still validates or secures the answer (the “or” case). Direct adaptations of prior methods estimate only $ATE_{nec}(1)$, which is logically insufficient to detect disjunctive contributions and thus systematically miscounts true-thinking steps.

Our evaluation reveals **true-thinking and decorative thinking steps are interleaved in a CoT**: while a sparse set of true-thinking steps directly influence the model’s predictions, others tend to act as decorative reasoning with negligible causal impact and are not truly used by models when computing their answer (Section 6). Additionally, we find **LLMs’ self-verification steps can be decorative without truly checking their solution** (Section 6.1). Beyond empirical evidence, we identify a mechanistic basis for this phenomenon: **whether an LLM internally performs a step verbalized in**

108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

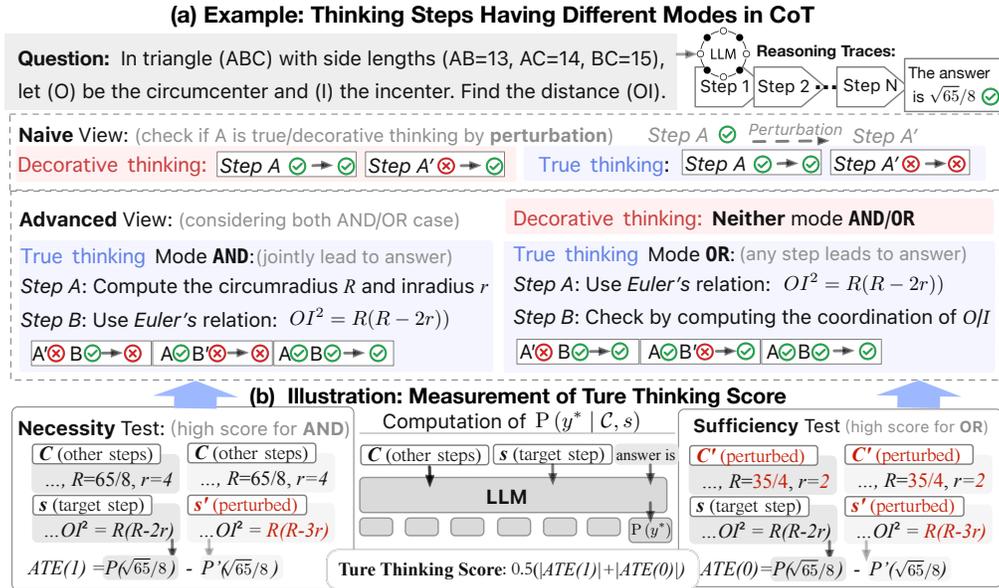


Figure 2: (a) Illustration of different modes in thinking steps within chain-of-thought (CoT) reasoning. Contrary to the naive view that a step’s faithfulness depends solely on whether perturbing it directly changes the final result, we show that the relationship is more nuanced. A true thinking step may operate in either an AND or OR mode when interacting with other steps. In both cases, such steps contribute meaningfully to the final answer. (b) Based on this understanding, we define the True Thinking Score, which jointly considers two complementary evaluations: the necessity test (high for AND-like steps) and the sufficiency test (high for OR-like steps).

CoT can be mediated by a TrueThinking direction in latent space (Section 7). Moving the hidden states of a step along this direction increases LLMs’ internal reliance on that step, whereas reversing it suppresses thinking over it. This also sheds light on a key limitation of existing faithfulness evaluations: they are hard to be directly verified, since doing so would require prior access to the model’s internal reasoning (Chen et al., 2025b). We propose that steering experiments offer an indirect testbed for validating such evaluation methods. Finally, we showcase that by steering along the TrueThinking direction, we causally induce the model to reason over decorative self-verification steps (Section 7.2).

Overall, we find that LLMs often narrate reasoning steps they do not actually execute internally. This gap fundamentally questions the efficiency of LLMs’ reasoning and undermines the practice of using verbalized rationales as a safety-monitoring signal (Baker et al., 2025). Our work shifts the focus from what models verbalize to what they think underneath, motivating future study that explains the mechanisms of CoT reasoning and develops training objectives that reward reliance on true-thinking steps.

2 RELATED WORK

Internal reasoning in LLMs’ latent space. Apart from relying on explicit CoT, LLMs also “think” internally across their layers. They can directly answer reasoning problems, sometimes even matching the performance of CoT-based prompting Ma et al. (2025). Prior works suggest that LLMs may solve certain tasks through internal *circuits* Yang et al. (2024); Marks et al. (2024); Prakash et al. (2025). Recent research showcases the *implicit* reasoning capabilities of LLMs that bypass explicit CoTs (Deng et al., 2023; Hao et al., 2024; Pfau et al., 2024; Goyal et al., 2023). The capability of internal reasoning of LLMs questions how much the model truly relies on each step verbalized in their CoTs. We study this gap by introducing a causal framework to evaluate each step in CoT.

Steering vectors in LLMs. Steering directions in latent space have been widely studied and have been found to mediate model’s behaviors/ perception in many aspects Von Rütte et al. (2024); Turner et al. (2023); Tigges et al. (2023); Li et al. (2023); Marks & Tegmark (2023). In terms of reasoning, past works have found steering vectors that can be used to control the strength of reasoning, e.g., longer or shorter CoT (Tang et al., 2025; Sun et al., 2025a; Chen et al., 2025a; Sheng et al., 2025) or different reasoning styles in CoT (Venhoff et al., 2025). We are the first to reveal that LLMs’ internal thinking process can also be mediated by steering vectors.

162 **Evaluating the faithfulness of reasoning traces.** Many works evaluate the faithfulness of
163 reasoning traces, but most treat the CoT as a whole, offering suggestive evidence without step-level
164 analysis. Existing methods fall into three categories: (1) Hint-based evaluation (Chen et al., 2025b;
165 Arcuschin et al., 2025; Chua & Evans, 2025; Turpin et al., 2023), where hints are injected into
166 unsolved questions and a faithful CoT should acknowledge them. These setups are not generalizable
167 and cannot assess individual steps. (2) Perturbation-based evaluation (Gao, 2023; Lanham et al.,
168 2023), where errors are injected into a step and the continuation is resampled. If the final answer
169 is unchanged, the CoT is deemed unfaithful. Yet this is unreliable, as the model may detect and
170 correct the error later in resampled steps (Yee et al., 2024). (3) Early-exit evaluation (Lanham et al.,
171 2023; Tanneru et al., 2024; Wang et al., 2025), where cues prompt the model to answer after an
172 intermediate step. A correct early answer suggests subsequent steps may be decorative, but this
173 ignores cases where later steps are faithfully used for e.g., self-verification or alternative solution,
174 etc. Overall, existing work has primarily provided suggestive, case-based evidence that CoTs are
175 not always faithful to the model’s internal thinking without looking into step-wise analysis. There
176 lacks a framework formally evaluating whether LLMs truly think over a step verbalized in CoTs
177 in their latent space. See Appendix C for more detailed related work.

178 3 MEASURING STEP-WISE CAUSALITY FOR FAITHFULNESS IN REASONING

179 Faithfulness in CoT is defined *with respect to a target*, typically the model’s predicted answer. A lack
180 of faithfulness arises when the model claims to rely on steps A, B, and C in its CoT, but internally
181 disregards them (instead, e.g., relying on other shortcuts or biases (Turpin et al., 2023; Arcuschin
182 et al., 2025) to compute answers). In this case, those steps make no causal contribution to the
183 prediction. Formally, we quantify the **causal contribution** of each reasoning step s in the CoT to
184 the final answer y^* , which serves as the basis for determining its faithfulness. A step with genuine
185 causal impact is a **true-thinking step**, where the model indeed internally thinks through s in order
186 to produce y^* . By contrast, a step with no causal impact is a **decorative-thinking step**, where the
187 model merely verbalizes a line of reasoning without using it internally.

188 **Notation and Setup.** We adopt notation following Rubin (1974); Pearl (2009). Let the input question
189 be q , and let the model’s full chain of thought (CoT) for q be $\mathcal{C}^* = (s_1, s_2, \dots, s_n)$, where each s_i
190 denotes a reasoning step. At the current step s under evaluation (we omit the index i and directly
191 use s for simplicity), we define the **context** as its preceding steps, i.e., $\mathcal{C} = (s_1, s_2, \dots, s_{i-1})$. To
192 probe the model’s current prediction after any partial reasoning trace, we use early-exit answering
193 by appending a standardized cue: `The final result is .` This approach, following Lanham
194 et al. (2023); Fu et al. (2025); Yang et al. (2025); Tanneru et al. (2024); Bogdan et al. (2025), reliably
195 elicits the model’s intermediate answer given the question q and reasoning prefix (\mathcal{C}, s) . Let $f(q, \mathcal{C}, s)$
196 denote the model’s early-exit prediction after processing q with context \mathcal{C} and step s . The *reference*
197 *prediction* under the full reasoning trace is then defined as $y^* := f(q, \mathcal{C}^*)$, representing the model’s
198 final answer when all steps in the full CoT are intact.

199 **Indicator Variables.** We introduce the following binary random variables to formalize interventions
200 on reasoning steps: (1) **Context indicator** $\mathbf{C} \in \{0, 1\}$: $\mathbf{C}=1$ indicates an intact context (the original
201 prefix \mathcal{C}); $\mathbf{C}=0$ indicates a perturbed context in which all preceding steps are replaced by perturbed
202 versions. We write $c \in \{0, 1\}$ for a specific realization of \mathbf{C} . (2) **Step toggle** $\mathbf{X} \in \{0, 1\}$: $\mathbf{X}=1$ is
203 the original step s ; $\mathbf{X}=0$ replaces it with a perturbed version s' . (3) **Outcome indicator** $\mathbf{Y} \in \{0, 1\}$:
204 Given $f(q, \mathcal{C}, s)$, we define $\mathbf{Y} := \mathbf{1}\{f(q, \mathcal{C}, s) = y^*\}$, which measures whether the model’s early-exit
205 prediction under the given intervention matches the full-CoT reference outcome.

206 **Perturbation Procedure.** To isolate the causal effect of each reasoning step, we create perturbed
207 versions of steps and contexts by introducing *small random numerical offsets* to quantities appearing
208 in the reasoning text (Lanham et al., 2023; Gao, 2023). These perturbations are minimal and preserve
209 grammatical and semantic structure, ensuring that the modified step remains coherent but subtly
210 altered. Additional implementation details are provided in Appendix B.

211 3.1 CONTEXT-BASED AVERAGE TREATMENT EFFECT

212
213 The Average Treatment Effect (ATE) (Rubin, 1974; Pearl, 2009) quantifies the causal effect of a
214 binary intervention $\mathbf{X} \in \{0, 1\}$ on an outcome \mathbf{Y} via Pearl’s $\text{do}(\cdot)$ operator:

$$215 \text{ATE} = S_1 - S_0, \quad S_{x \in \{0,1\}} := \Pr(\mathbf{Y}=1 \mid \text{do}(\mathbf{X}=x)). \quad (1)$$

To evaluate the causal contribution of a reasoning step s , we condition on a *context setting* $\mathbf{C} \in \{0, 1\}$ and define a *context-based ATE*:

$$\text{ATE}(c) = S_1(c) - S_0(c), \quad S_{x \in \{0,1\}}(c) := \Pr(\mathbf{Y}=1 \mid \mathbf{C}=c, \text{do}(\mathbf{X}=x)), \quad (2)$$

where $c \in \{0, 1\}$ specifies the context regime and \mathbf{X} toggles using the intact step s versus its perturbed counterpart s' . We consider two regimes: a) *Intact context* ($\mathbf{C}=1$): the original prefix \mathcal{C} preceding s is kept as generated; b) *Perturbed context* ($\mathbf{C}=0$): all steps in \mathcal{C} are minimally perturbed (e.g., by small numeric offsets), weakening associations between s and other steps so that the effect of s can be isolated.

Scoring the outcome. Let $y^* := f(q, \mathcal{C}^*)$ be the model’s reference answer obtained via *early-exit* on the full CoT, and let $f(q, \mathcal{C}, s)$ denote the early-exit prediction after a given (\mathcal{C}, s) . Instead of the binary $\mathbf{Y} = \mathbf{1}\{f(\cdot) = y^*\}$, we use the model’s confidence for the event y^* :

$$\Pr(\mathbf{Y}=1 \mid \cdot) \equiv \Pr(y^* \mid q, \mathcal{C}, s),$$

giving $S_x(c)$ a probabilistic (confidence-based) interpretation.

Interpreting $\text{ATE}_{\text{nec}}(1)$ vs. $\text{ATE}_{\text{suf}}(0)$. Conditioning on \mathbf{C} allows us to distinguish two complementary notions of causal relevance:

- **Necessity under intact context** ($\text{ATE}_{\text{nec}}(1)$).

$$\text{ATE}_{\text{nec}}(1) = \Pr(y^* \mid \mathbf{C}=1, \text{do}(\mathbf{X}=1)) - \Pr(y^* \mid \mathbf{C}=1, \text{do}(\mathbf{X}=0)).$$

This tests whether s is *needed* given the full, supportive context \mathcal{C} . A low $\text{ATE}_{\text{nec}}(1)$ indicates that removing s does not harm performance when other steps remain intact—what prior measures often label as “unnecessary.” However, this does *not* imply s is unfaithful; it may be redundant because other steps already suffice (an “OR” relation).

- **Sufficiency under perturbed context** ($\text{ATE}_{\text{suf}}(0)$).

$$\text{ATE}_{\text{suf}}(0) = \Pr(y^* \mid \mathbf{C}=0, \text{do}(\mathbf{X}=1)) - \Pr(y^* \mid \mathbf{C}=0, \text{do}(\mathbf{X}=0)).$$

This asks whether s can *on its own*—i.e., with weakened support from \mathcal{C} —drive the model toward y^* . A high $\text{ATE}_{\text{suf}}(0)$ suggests s is **sufficiently informative** to elicit the correct answer, capturing causal relevance even when s is not strictly necessary under the intact context.

Together, $\text{ATE}_{\text{nec}}(1)$ (necessity) and $\text{ATE}_{\text{suf}}(0)$ (sufficiency) provide a balanced view of **faithfulness**: a step can be causally meaningful by being necessary, sufficient, or both. The context perturbation operationalizes the “OR” case by dampening alternative pathways in \mathcal{C} , yielding a more reliable test of s ’s standalone impact.

True-Thinking Score (TTS). We define the faithfulness score of a step s with respect to the final result y^* as

$$\text{TTS}(s) = \frac{1}{2} (|S_1(1) - S_0(1)| + |S_1(0) - S_0(0)|). \quad (3)$$

A smaller $\text{TTS}(s)$ indicates that the step has little causal influence on the model’s prediction: perturbing or keeping it leads to almost the same result. Thus, that step is more likely to be *decorative* rather than *true thinking*. For each context setting c , we measure the unsigned $\text{ATE}(c)$, $|\text{ATE}(c)| = |S_1(c) - S_0(c)|$. The sign of $\text{ATE}(c)$ reflects whether the step is helpful or harmful (e.g., the step is actually wrong) overall, but we are interested in *how much* the model truly thinks through the step in its internal computation, regardless of direction. Taking the absolute value thus captures the magnitude of a step’s causal effect and provides a broader measure of its importance.

4 THE TRUETHINKING DIRECTION IN LLMs

In this section, we explain the methodology to extract a linear direction in the latent space of LLMs between *true thinking steps* (those with causal impact on the final answer) and *decorative thinking steps* (those with little or no impact). We call this latent vector **TrueThinking** direction. It can control whether the model truly thinks through a reasoning step and performs it internally. As illustrated in Figure 3, initially the model ignores the self-verification step s (“Wait, no, but ...”) and output

270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323

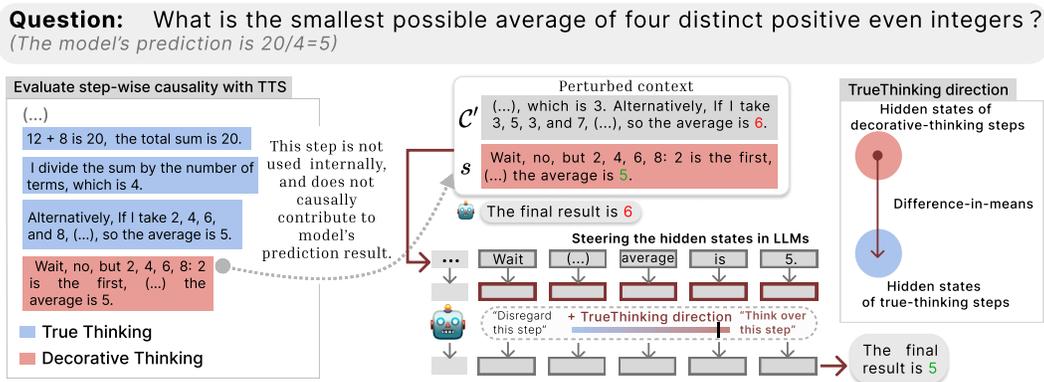


Figure 3: We uncover the TrueThinking direction in LLMs which is extracted as the difference between the mean hidden states of true-thinking steps and decorative-thinking steps. Steering the hidden states of each token in a step along this direction induces the model to truly think over that step in latent space.

the wrong answer 6 following the perturbed context. Steering the hidden states of step s along the TrueThinking direction makes the model truly think through that step and thus output the correct answer 5. We present detailed experiments in Section 7 and detail the methodology first in this section.

Formally, for each layer $l \in [1, L]$ in a Transformer-based (Vaswani et al., 2017) model, the hidden state for a token x_t in an input sequence x is updated with self-attention modules that associate x_t with tokens $x_{1:t}$ and a multi-layer perceptron: $h_t^l(x) = h_t^{l-1}(x) + \text{Attn}^l(x_t) + \text{MLP}^l(x_t)$. We focus on the residual stream activation $h^l(s_t)$ of the last token position t for a step s at a layer l . At a layer l , we collect the hidden states of the most representative true-thinking steps s_{TT} (where $\text{TTS}(s_{\text{TT}}) \geq \text{threshold } \alpha$) and decorative-thinking steps s_{DT} (where $\text{TTS}(s_{\text{DT}}) \leq \beta$). Following the difference-in-means approach (Marks & Tegmark, 2023; Arditì et al., 2024; Zhao et al., 2025), we compute the direction as the mean shift from $\mu_{\text{TT}}^l = \text{mean}(h^l(s_{\text{TT}}))$ to $\mu_{\text{DT}}^l = \text{mean}(h^l(s_{\text{DT}}))$ in the latent space.

$$v_{\text{TrueThinking}}^l = \mu_{\text{TT}}^l - \mu_{\text{DT}}^l. \quad (4)$$

This yields a steering vector that captures the model’s tendency to either sustain or truncate its reasoning process at that step. For steering at test time, we modify the residual stream for the hidden state of a test step in the example by using activation addition at a single layer l , i.e., $\bar{h}^l = h^l + v_{\text{TrueThinking}}^l$ to all tokens in the step. [We emphasize that thresholding is used only to select the most representative steps, while TTS itself is a continuous measure. As shown in Appendix G, steps with higher TTS scores tend to encode more evident true-thinking behavior.]

5 EXPERIMENTAL SETUP

Models. We conduct experiments on three different families of open-source reasoning models that have strong reasoning abilities and can generate long CoTs. For Qwen-2.5-7B and Llama-3.1-8B, we use the version finetuned on samples generated by Deepseek-R1 (Guo et al., 2025), i.e., Deepseek-R1-Distill-Qwen-7B¹ and Deepseek-R1-Distill-Llama-8B². We also experiment with Nemotron-1.5B³. These models can generate very long CoTs at test time. We use the maximum sequence length per model to avoid cut-off of reasoning traces during generation. We use greedy decoding for reproducibility and use the default prompting template for reasoning.

Data. We evaluate on three math reasoning benchmarks: (i) **AMC** (American Mathematics Competitions), (ii) **AIME** (American Invitational Mathematics Examination) from 2020–2024, and (iii) the **MATH** dataset (Hendrycks et al., 2021). For experiments in Section 7, we randomly sample 30% of cases as a heldout test set, 10 % of cases as validation set, and 60 % of cases as a training set to extract TrueThinking direction so that we can ensure our direction does not encode any information implying the answer of test cases. To compute the TrueThinking direction (explained in Section 4,

¹<https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B>

²<https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-8B>

³<https://huggingface.co/nvidia/OpenReasoning-Nemotron-1.5B>

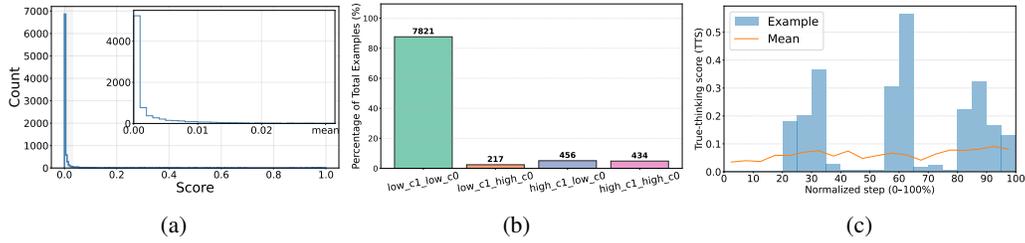


Figure 4: (a) The dataset-level distribution of the TTS score; (b) The distribution for $ATE_{nec}(1)$ and $ATE_{sur}(0)$ where *low* means $ATE(\cdot)$ is below mean and *high* means $ATE(\cdot)$ is above mean; (c) An example CoT case for TTS and the average TTS at different step percentile (normalized).

for all tested models we set the threshold $\alpha=0.9$ for s_{TT} to select the most representative true-thinking steps, while $\beta=0$ for selecting the most decorative steps s_{DT} , which means perturbing those steps does not change model’s confidence at all. Further ablation study is shown in Appendix G.

6 EVALUATION RESULTS OF STEP-WISE CAUSALITY IN CoT

In this section, we present evaluation results for the TTS score, which measures the extent to which the model truly reasons through each step internally. Recent reasoning models often produce long CoTs with many intermediate steps, incurring significant computational cost. We show that not each of these steps is truly used by the model in its internal reasoning process.

The distribution of TTS is long-tailed. As shown in Figure 4a, we find most steps have low scores, while only a few have very high scores. For example, as shown in Figure 4a on the AIME dataset of Qwen-2.5, the mean is around 0.03. Only 6.4% of CoT steps achieve a TTS greater than 0.3, and merely 2.3% exceed 0.7. This suggests that only a handful of verbalized steps in CoT are critical and faithfully followed by the model, whereas many others may not reliably reflect the model’s true inner thinking. Section 7 provides causal evidence to justify TTS. The ablation study in Appendix G further shows that, despite the long-tailed distribution, higher TTS indeed indicates that a step is more faithfully engaged in the model’s internal reasoning process. Additionally, our experimental results highlight the importance of evaluating both when the context is intact and when it is perturbed. In Figure 4b, we observe disparities between $ATE_{nec}(1)$ and $ATE_{sur}(0)$ for the same step, for example, cases where $ATE_{nec}(1)$ is low while $ATE_{sur}(0)$ is high. This indicates that solely relying on the score under an intact/perturbed context can miss potential true-thinking steps. We confirm the same pattern across datasets (see Appendix D), and steering experiments in Section 7 further show that the evaluation method that is only based on $c=1$ cases is unreliable. We observe similar results on Commonsense reasoning in Appendix E.

True-thinking steps and decorative-thinking steps are interleaved in a CoT. Figure 4c illustrates that steps with high TTS scores can appear at different positions, though later steps are on average more likely to be true-thinking with higher TTS. These results indicate that labeling an entire CoT as either unfaithful post-rationalization or faithful computation (Emmons et al., 2025) is overly coarse. They also raise concerns about the reliability of monitoring LLMs by inspecting CoT (Baker et al., 2025), since individual steps may not always reflect the model’s true internal reasoning or be performed internally at all. Finally, our results suggest that **task difficulty does not necessarily lead to more faithful reasoning**: even on the AIME dataset that challenges recent models (Sun et al., 2025b), LLMs still produce many decorative-thinking steps in CoT. The distribution of low TTS steps on AIME mirrors that of simpler math datasets (Appendix D), challenging the common hypothesis that LLMs tend to produce more faithful reasoning on harder problems (Emmons et al., 2025; Wang et al., 2025).

6.1 SELF-VERIFICATION STEPS CAN BE DECORATIVE

We leverage our defined TTS score to evaluate whether LLMs are truly thinking at self-verification steps (often known as “aha moments”). Self-verification steps are often seen in recent LLMs’ CoT, e.g., “Wait, let me recompute...”, which can help them achieve stronger reasoning performance (Guo et al., 2025; Muennighoff et al., 2025; Snell et al., 2024; Jaech et al., 2024). However, our findings suggest that in some cases the model may appear to self-verify in CoT, while internally it does not actually perform those steps and they make little contribution to its computation.

To identify decorative-thinking self-verification $\mathcal{S}_{\text{DT-self-verify}}$, we scan the self-verification steps and compute TTS. We define $\mathcal{S}_{\text{DT-self-verify}} = (s_1, s_2, \dots, s_n)$ where $\text{TTS}(s_i)$ is smaller than a threshold β . Notably, we observe cases where self-verification steps have near-zero TTS (see an instance in Figure 13 in the Appendix). For example, around 12% of the self-verification steps for Qwen-2.5 have TTS lower than 0.005, while 21% for Nemotron. We also find that perturbing the context steps before $\mathcal{S}_{\text{DT-self-verify}}$ can always flip the model’s initial correct answers to wrong ones, though $\mathcal{S}_{\text{DT-self-verify}}$ may contain ample information to lead the model to correct answers. Overall, those self-verification steps contribute minimally to the model’s computation of its answer. We further demonstrate that we can force the model to truly self-check its solution using steering vectors in Section 7.2, and provide more examples with different reasoning behaviors in Appendix H.

7 TRUE THINKING CAN BE MEDIATED BY A STEERING DIRECTION

In this section, we empirically show that for LLMs, **whether to truly think through a verbalized reasoning step or disregard it internally can be mediated by a steering direction in latent space** (i.e., our identified TrueThinking direction in Section 4). We first explain two causal tests designed to reveal such internal behaviors of LLMs, and then present the main experimental findings in Section 7.1.

Causal tests. We design two steering tasks to investigate the mechanism of LLMs’ thinking in CoT. **Engagement Test: Can steering make the model think through a step in CoT it normally ignores?** We consider cases where $f(q, \mathcal{C}) = y_{GT}$ and $f(q, \mathcal{C}, s') = y_{GT}$. Namely, the model obtains the ground-truth answer y_{GT} without the step s and with the perturbed s (i.e., s'). If we apply the direction $v_{\text{TrueThinking}}^l$ to the hidden state of s' , and the model’s correct answer flips to an incorrect one ($f^{+v_{\text{TrueThinking}}^l}(q, \mathcal{C}, s') \neq y_{GT}$), this indicates that the intervention has forced the model to reason over s' , following the errors injected into s' . **Disengagement Test: Can steering in the reverse direction make the model disregard a step internally?** Now consider cases where the model predicts the correct answer before step s , i.e., $f(q, \mathcal{C}) = y_{GT}$, but including a perturbed step s' causes it to fail: $f(q, \mathcal{C}, s') \neq y_{GT}$. If applying $-v_{\text{TrueThinking}}^l$ to s' flips the wrong answer to the correct answer ($f^{-v_{\text{TrueThinking}}^l}(q, \mathcal{C}, s') = y_{GT}$), then the intervention has made the model disregard the step s' .

Comparison baselines. As baselines, we consider three approaches for layer-wise intervention. (1) DropStep: adapted from prior work (Tanneru et al., 2024; Lanham et al., 2023; Bogdan et al., 2025; Wang et al., 2025), this method compares prediction confidence before and after appending step s , i.e., $P(y_{GT}|q, \mathcal{C}, o) - P(y_{GT}|q, \mathcal{C})$, where a larger difference indicates true-thinking steps; the identified steps are then used to extract a steering direction following the same method in Section 4. (2) Attention scaling: we directly scale the attention weights of the tokens of step s at a layer during inference time, with $\text{scale}=100$ encouraging the model to think through the step and $\text{scale}=0$ suppressing it. (3) Random steering vector: we generate a random vector with the same dimensionality and norm as the TrueThinking direction to test whether our identified direction encodes meaningful information.

7.1 RESULTS

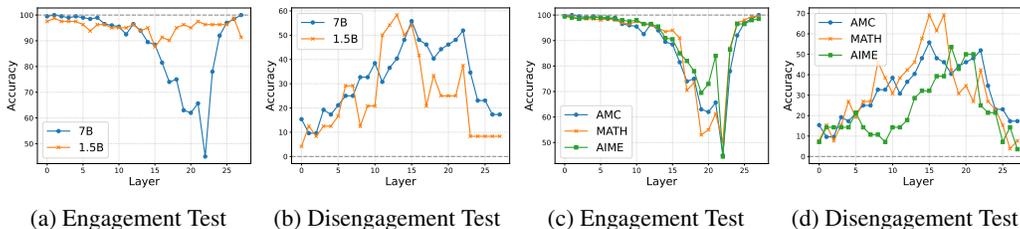
LLMs encode a steerable latent signal of “thinking”. We follow the method detailed in Section 4 for steering, and our results show that a simple linear TrueThinking direction mediates whether LLMs truly reason over a verbalized step. As shown in Table 1, steering with the (reverse) TrueThinking direction reliably flips predictions in both tests. In the Disengagement Test, it effectively prevents the model from using the perturbed step s' , with effects far stronger than those of random vectors. This shows that suppression of step use with the reverse TrueThinking direction in the Disengagement Test arises from a meaningful signal rather than added noise, confirming that the TrueThinking direction captures a genuine internal representation of *thinking*. We also compare models with different sizes in the same model family. Similar patterns hold for the much smaller Qwen-2.5-1.5B model (Figure 5a and Figure 5b) and larger 14B model (Figure 11 in Appendix F). Steering along the TrueThinking direction (to induce the step use in LLMs’ internal reasoning) is weaker than in Qwen-2.5-7B, whereas the results in the Disengagement Test are comparable.

On the other hand, our experiments across datasets show that **the latent signal controlling whether a step engages in reasoning is universal**. As seen in Table 1, the TrueThinking direction extracted on AMC generalizes well to other datasets across all models, indicating a model-internal mechanism of thinking rather than a dataset-specific artifact. For instance, in the Qwen model, layer 15-22 consistently yield the strongest intervention performance across all three datasets (Figure 5c-5d), suggesting these intermediate layers concentrate latent reasoning.

432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448

Dataset / Method	Attention scaling		Random vector		DropStep		Ours	
	ET	DT	ET	DT	ET	DT	ET	DT
<i>AMC</i>								
Qwen-2.5	6.2	25.0	4.0	26.9	1.5	28.6	55.0	55.7
Llama-3.1	24.8	20.5	3.5	20.6	10.5	32.4	17.6	35.3
Nemotron	5.1	27.2	4.5	45.5	9.0	45.4	35.7	54.5
<i>MATH</i>								
Qwen-2.5	10.0	23.9	2.0	30.2	2.5	17.7	49.8	69.2
Llama-3.1	7.5	35.4	5.0	47.9	11.0	52.1	14.0	54.2
Nemotron	21.7	42.7	21.5	44.6	6.5	45.1	59.5	56.3
<i>AIME</i>								
Qwen-2.5	9.3	25.0	1.5	21.4	1.5	14.3	55.5	53.6
Llama-3.1	6.3	35.2	2.5	29.4	5.0	41.1	38.0	47.1
Nemotron	12.0	70.6	6.5	76.5	4.5	79.5	39.0	91.2

449 Table 1: Top-1 flip rate among all layers (%) \uparrow in the Engagement Test (ET) and the
450 Disengagement Test (DT). We use flip rate as the metric, measuring how often steering changes the
451 model’s initial prediction. AMC dataset is in-domain evaluation where TrueThinking directions are
452 extracted, while the other two datasets are for out-of-domain evaluation.



459
460
461 Figure 5: Layer-wise results of steering with the TrueThinking vector. In the Engagement Test,
462 stronger intervention is reflected by lower accuracy (more right→wrong flips); In the
463 Disengagement Test, by higher accuracy (more wrong→right flips). Figures (a–b): layer-wise
464 results on AMC for DeepSeek-R1-Distill-Qwen-7B and its 1.5B variant under the Engagement Test
465 and the Disengagement Test. Figures (c–d): cross-domain results, where the TrueThinking direction
466 is extracted on AMC and applied to MATH and AIME.

467
468 **Causal steering provides a testbed to validate faithfulness metrics.** Despite extensive work on
469 evaluating the faithfulness of reasoning traces, there is no framework to verify these metrics, since
470 the ground truth of whether a model truly *thinks* through a step is inherently inaccessible (Chen
471 et al., 2025b). We propose causal steering as an indirect validation framework: if a metric identifies
472 meaningful steps, then the directions it extracts should causally mediate whether the model engages
473 with a step in its internal reasoning. Empirically, steering directions derived from our TTS score
474 produce stronger and more consistent intervention effects than DropStep of past works (Tanneru et al.,
475 2024; Lanham et al., 2023; Bogdan et al., 2025; Wang et al., 2025). We also conduct an ablation study
476 showing that averaging over ATE(c) in Eq. 3 leads to better steering performance in Appendix G.2.

477
478 **Steering with the TrueThinking direction mediates LLMs’ attention.** We find that True-
479 Thinking direction may control the model’s internal reasoning process by reallocating atten-
480 tion among tokens. In Disengagement Test, steering in the reverse TrueThinking direction (for example,
481 when applied at Layer 17) reduces attention to the perturbed tokens, as shown in Figures 6a and 6b,
482 causing the model to largely disregard them. Similarly, steering along the TrueThinking direction in-
483 creases attention to the step (see examples in Figure 8a and Figure 8b of Appendix). Overall, when
484 steering along the TrueThinking direction, the attention of the steered tokens will increase on average
485 by 193% and steering along the reverse direction will decrease the attention by 32.3% across the layers. In comparison, directly scaling
attention on step tokens in a layer does not always yield noticeable effects. As shown in Table 1, in the
Disengagement Test, masking attention (i.e., setting coefficients to 0) at a layer can partially flip

486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539

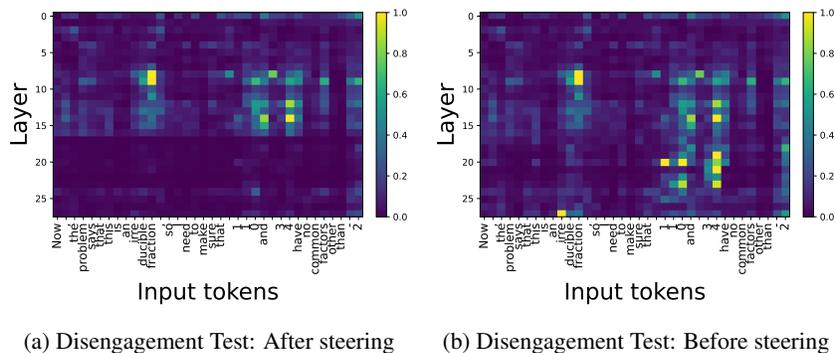


Figure 6: Normalized attention scores of the step in the Engagement Test and the Disengagement Test before and after steering. Applying the reverse TrueThinking direction decreases the model’s attention.

answers, but in the Engagement Test its impact is weak, suggesting that attention alone does not drive/ suppress reasoning. We hypothesize that LLMs employ a directional reasoning *circuit* (Marks et al., 2024; Prakash et al., 2025), where the model first decides whether to engage in reasoning for a step and only then modulates attention, which may be irreversible through direct attention scaling. We leave understanding the relation between attention and the reasoning mechanism for future work.

7.2 STEERING DECORATIVE SELF-VERIFICATION STEPS

As shown in Section 6.1, self-verification in CoT can be decorative and not really engaged with LLMs’ internal reasoning. We investigate whether steering along the TrueThinking direction can force the model to truly think through $\mathcal{S}_{\text{DT-self-verify}}$ and thus restore the correct answer. Specifically, we study cases where the model produces the correct answer after $\mathcal{S}_{\text{DT-self-verify}}$, namely, $f(q, \mathcal{C}, \mathcal{S}_{\text{DT-self-verify}}) = y_{\text{GT}}$. We then perturb \mathcal{C} to obtain \mathcal{C}' such that $f(q, \mathcal{C}', \mathcal{S}_{\text{DT-self-verify}}) \neq y_{\text{GT}}$. Next, following Section 4, we apply the TrueThinking direction to the tokens in $\mathcal{S}_{\text{DT-self-verify}}$, encouraging the model to genuinely think through $\mathcal{S}_{\text{DT-self-verify}}$ and measuring how much this steering restores the correct answer.

We find that steering along the TrueThinking direction can at best reverse 52% of the unfaithful self-verification steps in CoT (layer-wise results shown in Figure 7). Remarkably, on the tested Deepseek-distilled-R1-Qwen-7B model, the layer with the strongest intervention effect aligns with the layer identified in Section 7, suggesting that certain layers may play a special role in determining whether reasoning steps are engaged in the model’s internal computation. Beyond interpretability, this finding points toward applications in efficient reasoning: the discovered TrueThinking direction could be leveraged to maximize token budget utilization, ensuring the model truly reasons over each generated token rather than producing superficially coherent but ungrounded steps.

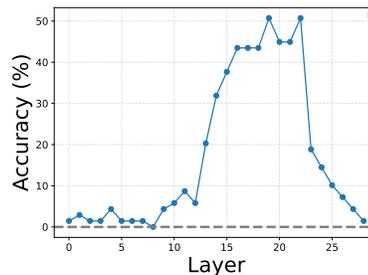


Figure 7: Performance after steering the model to truly think over the self-verification part, where initially the accuracy is zero.

8 CONCLUSIONS

We propose a step-wise causality framework to evaluate CoT faithfulness, revealing that *true-thinking* and *decorative-thinking* steps are interleaved: only a small subset are *true-thinking* that causally influence predictions, whereas most are *decorative-thinking* that merely create the appearance of reasoning and have minimal causal impact on predictions. Mechanistically, we demonstrate that whether a reasoning step in CoT contributes to a model’s computation can be controlled by a TrueThinking direction, enabling causal steering for the model to either follow or disregard that step in its internal thinking process. Steering tests can also provide an indirect validation testbed for evaluating faithfulness metrics. Overall, our findings show that many steps in CoT do not faithfully reflect an LLM’s internal thinking: models may verbalize reasoning they do not actually perform. This raises concerns about both the efficiency of LLMs’ reasoning and the reliability of relying on CoT to monitor LLMs for safety.

540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593

REFERENCES

- Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthoooran Rajamanoharan, Neel Nanda, and Arthur Conmy. Chain-of-thought reasoning in the wild is not always faithful. *arXiv preprint arXiv:2503.08679*, 2025.
- Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.
- Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *arXiv preprint arXiv:2503.11926*, 2025.
- Paul C Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. Thought anchors: Which llm reasoning steps matter? *arXiv preprint arXiv:2506.19143*, 2025.
- Runjin Chen, Zhenyu Zhang, Junyuan Hong, Souvik Kundu, and Zhangyang Wang. Seal: Steerable reasoning calibration of large language models for free. *arXiv preprint arXiv:2504.07986*, 2025a. doi: 10.48550/arXiv.2504.07986. URL <https://arxiv.org/abs/2504.07986>.
- Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. Reasoning models don’t always say what they think. *arXiv preprint arXiv:2505.05410*, 2025b.
- James Chua and Owain Evans. Are deepseek r1 and other reasoning models more faithful? *arXiv preprint arXiv:2501.08156*, 2025.
- Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*, 2023. doi: 10.48550/arXiv.2311.01460.
- Scott Emmons, Erik Jenner, David K Elson, Rif A Saurous, Senthoooran Rajamanoharan, Heng Chen, Irhum Shafkat, and Rohin Shah. When chain of thought is necessary, language models struggle to evade monitors. *arXiv preprint arXiv:2507.05246*, 2025.
- Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. *arXiv preprint arXiv:2508.15260*, 2025.
- Leo Gao. Shapley value attribution in chain of thought. URL <https://www.lesswrong.com/posts/FX5JmftqL2j6K8dn4/shapley-value-attribution-in-chain-of-thought>, 2023.
- Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. Roscoe: A suite of metrics for scoring step-by-step reasoning. *arXiv preprint arXiv:2212.07919*, 2022.
- Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*, 2023.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E. Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

594 Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. C3ot: Generating shorter chain-of-thought with-
595 out compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*,
596 volume 39, pp. 24312–24320, 2025.

597

598 Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, Mark
599 Chen, Alan Cooney, Allan Dafoe, Anca Dragan, et al. Chain of thought monitorability: A new and
600 fragile opportunity for ai safety. *arXiv preprint arXiv:2507.11473*, 2025.

601

602 Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernan-
603 dez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in
604 chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

605

606 Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time
607 intervention: Eliciting truthful answers from a language model. *Advances in Neural Information
608 Processing Systems*, 36:41451–41530, 2023.

609

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

612

613 Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language
614 model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

615

616 Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse
617 feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv
618 preprint arXiv:2403.19647*, 2024.

619

620 Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke
621 Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time
622 scaling. *arXiv preprint arXiv:2501.19393*, 2025.

623

624 Judea Pearl. *Causality*. Cambridge university press, 2009.

625

626 Jacob Pfau, William Merrill, and Samuel R Bowman. Let’s think dot by dot: Hidden computation in
627 transformer language models. *arXiv preprint arXiv:2404.15758*, 2024.

628

629 Nikhil Prakash, Natalie Shapira, Arnab Sen Sharma, Christoph Riedl, Yonatan Belinkov, Tamar Rott
630 Shaham, David Bau, and Atticus Geiger. Language models use lookbacks to track beliefs. *arXiv
631 preprint arXiv:2505.14685*, 2025.

632

633 Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies.
634 *Journal of educational Psychology*, 66(5):688, 1974.

635

636 Leheng Sheng, An Zhang, Zijian Wu, Weixiang Zhao, Changshuo Shen, Yi Zhang, Xiang Wang,
637 and Tat-Seng Chua. On reasoning strength planning in large reasoning models. *arXiv preprint
638 arXiv:2506.08390*, 2025. doi: 10.48550/arXiv.2506.08390. URL [https://arxiv.org/abs/
639 2506.08390](https://arxiv.org/abs/2506.08390).

640

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

643

644 Jiwon Song, Dongwon Jo, Yulhwa Kim, and Jae-Joon Kim. Reasoning path compression: Com-
645 pressing generation trajectories for efficient llm reasoning. *arXiv preprint arXiv:2505.13866*,
646 2025.

647

648 Chung-En Sun, Ge Yan, and Tsui-Wei Weng. Thinkedit: Interpretable weight editing to mitigate
649 overly short thinking in reasoning models. *arXiv preprint arXiv:2503.22048*, 2025a. doi: 10.
650 48550/arXiv.2503.22048. URL <https://arxiv.org/abs/2503.22048>.

651

652 Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, Zheng Liu, Zhongyuan
653 Wang, and Ji-Rong Wen. Challenging the boundaries of reasoning: An olympiad-level math
654 benchmark for large language models. *arXiv preprint arXiv:2503.21380*, 2025b.

648 Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question
649 answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of*
650 *the North American Chapter of the Association for Computational Linguistics: Human Language*
651 *Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019.

652 Xinyu Tang, Xiaolei Wang, Zhihao Lv, Yingqian Min, Wayne Xin Zhao, Binbin Hu, Ziqi Liu,
653 and Zhiqiang Zhang. Unlocking general long chain-of-thought reasoning capabilities of large
654 language models via representation engineering. *arXiv preprint arXiv:2503.11314*, 2025. doi:
655 10.48550/arXiv.2503.11314. URL <https://arxiv.org/abs/2503.11314>. ACL 2025.

656 Sree Harsha Tanneru, Dan Ley, Chirag Agarwal, and Himabindu Lakkaraju. On the hardness of
657 faithful chain-of-thought reasoning in large language models. *arXiv preprint arXiv:2406.10625*,
658 2024.

659 Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Linear representations of
660 sentiment in large language models. *arXiv preprint arXiv:2310.15154*, 2023.

661 Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini,
662 and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint*
663 *arXiv:2308.10248*, 2023.

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

667 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz
668 Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing*
669 *systems*, 30, 2017.

670 Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding
671 reasoning in thinking language models via steering vectors. *arXiv preprint arXiv:2506.18167*,
672 2025.

673 Dimitri Von Rütte, Sotiris Anagnostidis, Gregor Bachmann, and Thomas Hofmann. A language
674 model’s guide through latent space. *arXiv preprint arXiv:2402.14433*, 2024.

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

679 Tom Wollschläger, Jannes Elstner, Simon Geisler, Vincent Cohen-Addad, Stephan Günnemann,
680 and Johannes Gasteiger. The geometry of refusal in large language models: Concept cones and
681 representational independence. In *Forty-second International Conference on Machine Learning*,
682 2025.

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

685 Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Zheng Lin, Li Cao,
686 and Weiping Wang. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*,
687 2025.

688 Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language
689 models latently perform multi-hop reasoning? *arXiv preprint arXiv:2402.16837*, 2024.

690 Evelyn Yee, Alice Li, Chenyu Tang, Yeon Ho Jung, Ramamohan Paturi, and Leon Bergen. Dissocia-
691 tion of faithful and unfaithful reasoning in llms. *arXiv preprint arXiv:2405.15092*, 2024.

692 Xiangning Yu, Zhuohan Wang, Linyi Yang, Haoxuan Li, Anjie Liu, Xiao Xue, Jun Wang, and
693 Mengyue Yang. Causal sufficiency and necessity improves chain-of-thought reasoning. *arXiv*
694 *preprint arXiv:2506.09853*, 2025.

695 Jiachen Zhao, Jing Huang, Zhengxuan Wu, David Bau, and Weiyan Shi. Llms encode harmfulness
696 and refusal separately. *arXiv preprint arXiv:2507.11878*, 2025.

702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755

A LIMITATIONS

Our causal evaluation framework is inherently approximate. It is greedy in nature and may not capture all possible causal pathways, nor does it aim to reconstruct a complete causal graph of reasoning steps. Thus, it should be viewed as a probe that highlights representative *true-thinking* and *decorative-thinking* steps rather than a definitive oracle of internal reasoning. In addition, the TrueThinking direction we extract may not be optimal. We regard our findings as an existence proof that internal thinking can be mediated by steering directions, and we leave the development of more effective directions and a deeper understanding of their geometry to future work. We cannot experiment on larger frontier models due to limited computational resources, and our findings may therefore not fully generalize to those untested settings. Nonetheless, by demonstrating effectiveness across several accessible models, we establish a general evaluation framework for analyzing and interpreting the thinking process in CoT.

[Our TTS computation can be costly as it requires different runs. However, in this work, we do not aim to propose an efficient real-time detector, but in the first place, we need a theoretically sound way to reveal whether steps in CoT are faithful. Future work can leverage the TrueThinking direction to construct a latent monitor by comparing it with the hidden states.]

B IMPLEMENTATIONS

Perturbing reasoning steps. We treat sentences as distinct reasoning steps, as prior work has shown that each sentence can serve a different function within a reasoning trace (Bogdan et al., 2025). We follow prior work (Lanham et al., 2023; Gao, 2023) and add small random offsets (chosen from $[-3, -2, -1, 1, 2, 3]$) to the numbers in a reasoning step. This keeps the perturbation minimal so that the step remains largely unchanged in token length, wording, and underlying logic. We can therefore reasonably attribute any confidence changes caused by the perturbation to the model’s treatment of the original step. For steps that do not contain numerical values, we also follow prior work (Bogdan et al., 2025; Tanneru et al., 2024; Wang et al., 2025) by dropping them as a perturbation (i.e., applying $\text{do}(\mathbf{X} = 0)$) to measure the influence of those sentences. For perturbing context steps, we only change numerical values.

C MORE RELATED WORK

Detailed analysis of existing evaluation frameworks of faithfulness for CoT. Many recent works have sought to evaluate the faithfulness of reasoning traces. Most, however, focus on the CoT as a whole, providing suggestive evidence that the CoT is not faithful without analyzing each step. The existing evaluation methods can be summarized as,

- **Hint-based evaluation:** Most prior studies (Chen et al., 2025b; Arcuschin et al., 2025; Chua & Evans, 2025; Turpin et al., 2023) consider simple multiple-choice questions rather than complex reasoning tasks. Hints are injected into questions that the model initially failed to solve. A faithful CoT should explicitly acknowledge the use of hints in deriving the correct answer (Chua & Evans, 2025; Chen et al., 2025b). Relatedly, Arcuschin et al. (2025); Turpin et al. (2023) insert biasing features into questions and observe whether the model’s answer changes. If so, the CoT is deemed unfaithful, as the prediction is driven by bias in the prompt. Yet, those framework setups are not generalizable to practical reasoning problems, and cannot reveal the faithfulness of individual steps.
- **Perturbation-based evaluation:** Errors are injected into a correct reasoning step, and its following reasoning traces are resampled (Gao, 2023; Lanham et al., 2023). If the model’s predicted answer remains unchanged at last, the CoT is considered unfaithful since the error inserted was ignored. However, this criterion is unreliable: the model may instead detect and correct the error in later resampled steps. Yee et al. (2024) try to address this by manually reviewing self-correction steps, but such methods already assume that verbalized steps faithfully reflect the model’s computation as a priori.
- **Early-exit answering:** Early-exit cues are inserted after a reasoning step to test whether the model can already produce a correct answer (Lanham et al., 2023; Tanneru et al., 2024). A

756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

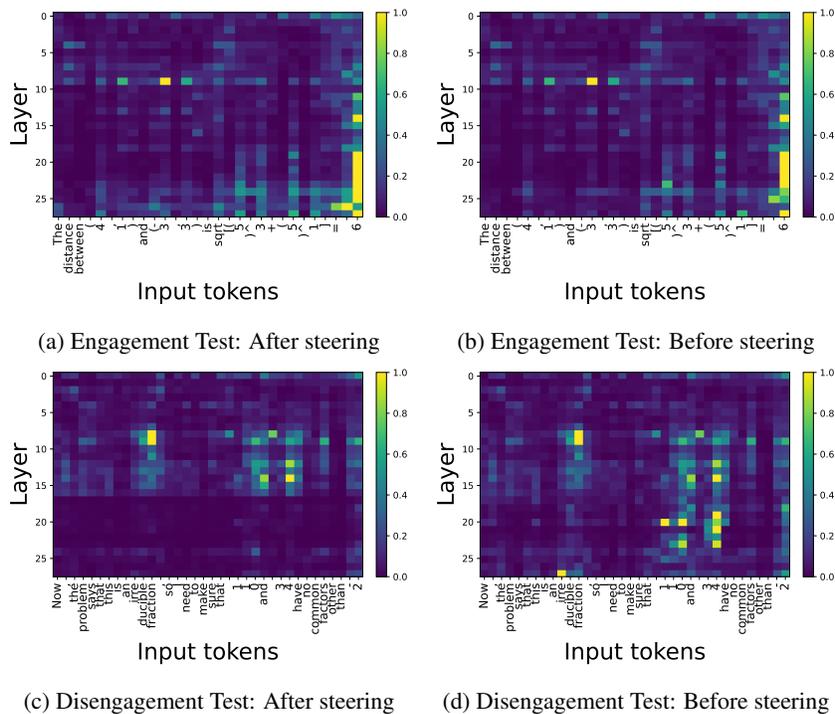


Figure 8: Normalized attention scores of the step in the Engagement Test and the Disengagement Test before and after steering. (a–b) Applying the TrueThinking direction to a step increases the model’s attention to it. (c–d) Applying the reverse TrueThinking direction decreases the model’s attention.

correct early-exit answer suggests the CoT may be unfaithful (Lanham et al., 2023) since further steps are unnecessary to model’s answer. Tanneru et al. (2024) further computes the change in the model’s answer confidence before and after each reasoning step when using early-exit answering. Yet unnecessary may not be equivalent to unfaithfulness. The fact that a model arrives at the correct answer early and maintains it does not necessarily imply that it ignores subsequent reasoning steps. This view overlooks important cases where the model continues to engage in those steps, for example, faithfully performing self-verification to consolidate or reinforce earlier predictions.

On the other hand, conceptually, CoTs have also been hypothesized as either *CoT-as-computation* or *CoT-as-rationalization* (Emmons et al., 2025). However, our analysis in Section 6 reveals a more nuanced picture: reasoning traces in CoT often interleave steps that genuinely drive computation with others that merely pretend reasoning.

CoT pruning. Our work is also broadly related to CoT compression (Xia et al., 2025; Song et al., 2025; Kang et al., 2025; Yu et al., 2025) and important steps analysis (Golovneva et al., 2022). However, those works are different from “faithfulness” as they focus on the semantics of verbalized steps for analysis and do not look into the underlying causality. For example, Xia et al. (2025) prune long CoTs into shorter sentences with equivalent meanings for LLMs without degrading performance. But that does not mean pruned steps do not causally contribute to the model’s answer. Our work may suggest similar results that there exist shorter alternative CoTs leading the model to get the same answers, but they do not solve whether a step truly engages with the model’s internal computation, and the internal mechanism of LLMs’ thinking process is still unclear. Yu et al. (2025) reframe CoT pruning into the Probability of Necessity and Sufficiency (PNS) framework (Pearl, 2009), but it focuses on estimating the necessity of each step using counterfactual intervention in a similar way to prior works (Lanham et al., 2023; Yee et al., 2024; Bogdan et al., 2025): They perturb an intermediate step while keeping all preceding steps intact, then resample the subsequent steps using a rollout model. The resulting reasoning trace is evaluated by a validator model to assess whether the initial step is necessary. Although they are framed into PNS, their measurement of sufficiency and necessity

810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863

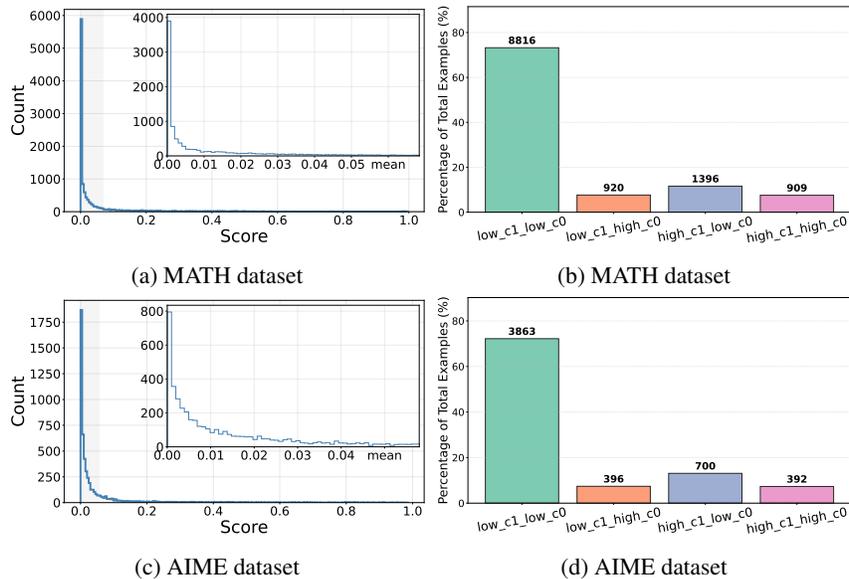


Figure 9: Distribution of TTS on different datasets.

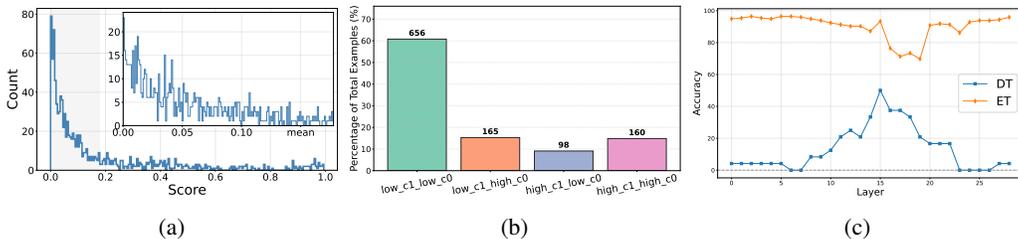


Figure 10: Results for Commonsense reasoning. (a) The dataset-level distribution of the TTS score; (b) The distribution for $ATE(c = 1)$ and $ATE(c = 0)$ where *low* means $ATE(\cdot)$ is below mean and *high* means $ATE(\cdot)$ is above mean; (c) Steering results on the Engagement Test and the Disengagement Test.

is restricted to contexts where the prior steps remain unchanged, which corresponds to $ATE(1)$ in our framework. From a faithfulness perspective, this neglects disjunctive cases where a step is not strictly necessary for producing the correct answer but still reflects genuine reasoning. They also neglect that the model can realize and correct the previously-perturbed step in the rollout steps (Yee et al., 2024).

D MORE EXPERIMENTAL RESULTS OF TTS

As shown in Figure 9, we observe a similar long-tail distribution of TTS across different datasets. Steps with larger TTS are sparse, while the most of steps have relatively low TTS. Additionally, as shown in Figure 9b and Figure 9d, on different datasets, we still observe that there exist ample examples where $ATE(c = 1)$ and $ATE(c = 0)$ of the same step are very different.

E RESULTS ON COMMONSENSE REASONING

We apply TTS and the TrueThinking direction to commonsense reasoning to test the generalizability of our framework to more generic reasoning cases. We use CommonsenseQA (Talmor et al., 2019) as our test data. Since commonsense reasoning is predominantly non-numeric, standard intervention schemes that rely on perturbing numerical values are not suitable in this setting. To apply interventions on such content, we prompt a separate large language model (GPT-5) to produce subtle, semantically

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

Question	Where would you find magazines alongside many other printed works?				
Options	A doctor	B ✓ bookstore	C market	D train station	E mortuary
Original rationale	... <u>bookstores</u> are designed to sell both books and periodicals...				
Perturbed rationale	... <u>train stations</u> are designed to sell both books and periodicals...				

Table 2: Example multiple-choice question with original vs. perturbed rationale.

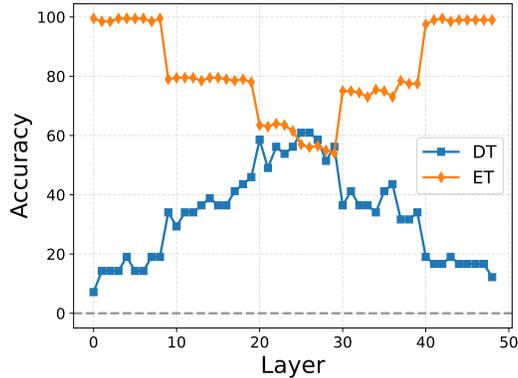


Figure 11: Results of Engagement Test and Disengagement Test on Qwen-14B.

coherent distortions of the initial reasoning step. This makes perturbation applicable to different reasoning scenarios. An example perturbed rationale is shown in Table 2.

As shown in Figure 10a and Figure 10b, the distribution of TTS on Commonsense reasoning is long-tailed, which is similar to the patterns observed on mathematical reasoning.

Furthermore, we conduct the causal tests (explained in Section 7) on the examples of Commonsense QA. Specifically, we use the same TrueThinking direction extracted from AMC dataset for steering (the same setting as Table 1). The steering results are shown in Figure 10c. Noticeably, the TrueThinking direction can still mediate the reasoning behaviors on CommonsenseQA when applied to the middle layers. This strongly indicates the generalizability of TrueThinking direction and suggests that LLMs encode a specific variable for true-thinking behaviors.

F STEERING RESULTS ON LARGER MODELS

Beyond the 1.5B and 7B models in Figure 5a and Figure 5b, we further demonstrate that the TrueThinking direction generalizes to a larger 14B model. As shown in Figure 11, we can similarly modulate the 14B model’s reasoning behavior, controlling whether it engages or disengages with reasoning steps by steering the hidden states in the middle layers along or against TrueThinking direction.

G ABLATION STUDY

G.1 THRESHOLD OF TTS FOR TRUE-THINKING STEPS

In this section, we ablate the threshold for selecting true-thinking steps when extracting the TrueThinking vector. Our goal is to better understand the scale of TTS, that is, how low a score may already indicate decorative thinking and how high a score reflects true thinking. We use steering performance as an indirect probe of how TTS correlates with the internal engagement of steps in reasoning.

When extracting steering directions with difference-in-means, the steps with zero TTS are treated as decorative-thinking steps (s_{DT}), while we use steps from different ranges of TTS as true-thinking steps (s_{TT}). As shown in Figure 12, when TrueThinking directions are computed from steps drawn from

increasingly higher TTS ranges, the steering effect grows roughly linearly. In contrast, using steps with TTS below 0.03 yields negligible steering, suggesting these steps are internally treated as decorative, similarly to those with zero TTS. Because the TrueThinking directions are computed as the difference in mean hidden states between true and decorative steps (Eq. 4), negligible steering effects mean the direction fails to capture the meaningful distinction between decorative and true thinking.

Overall, this analysis reveals an implicit decision boundary in the model’s internal space: while the distribution of TTS is very long-tailed and high-TTS steps are rare, larger TTS indeed corresponds to genuinely influential reasoning. We leave further in-depth study of the geometry (Wollschläger et al., 2025) of hidden states and TrueThinking directions in the latent space as future work.

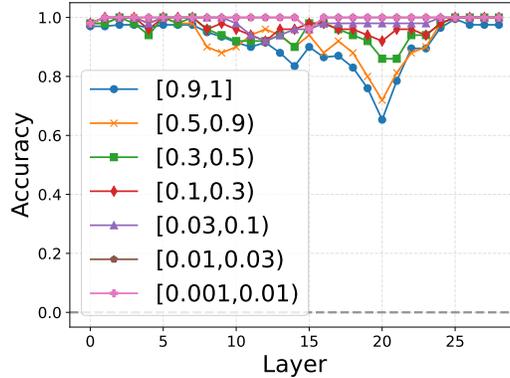


Figure 12: Test results of Nemotron on the Engagement Test where TrueThinking directions are extracted between examples with zero TTS (as decorative-thinking steps s_{DT}) and examples of different ranges of TTS (as true-thinking steps s_{TT}), and the lower accuracy means stronger steering effects.

G.2 AVERAGING OVER $ATE(c)$ FOR TTS

We compare using only $ATE(1)$ in TTS with the complete TTS to identify true-thinking steps. This slightly differs from the DropStep method in Table 3, which is adopted by past works (Tanneru et al., 2024; Lanham et al., 2023; Wang et al., 2025; Bogdan et al., 2025). DropStep defines $ATE(1)$ with step removal as the perturbation, i.e., $ATE(1)^{drop} = P(y^* | \mathcal{C}, s) - P(y^* | \mathcal{C})$. In contrast, here $ATE(1) = P(y^* | \mathcal{C}, s) - P(y^* | \mathcal{C}, s')$ using numerical perturbation by changing the numbers in step s . However, as shown in Table 3, incorporating $ATE(0)$ is still necessary, as the resulting TrueThinking direction achieves stronger intervention performance.

	$ ATE_{nec}(1) $ only		$\text{mean}(ATE_{nec}(1) , ATE_{suf}(0))$	
	A	B	A	B
Flip rate	10	41.6	38.0	62.5

Table 3: Results on MATH for steering with directions identified from true-thinking steps based on $ATE(1)$ and complete TTS.

972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

Problem: A square has a side length of 10 inches. Congruent isosceles right triangles are cut off each corner so that the resulting octagon has equal side lengths. How many inches are in the length of one side of the octagon? Express your answer as a decimal to the nearest hundredth.

CoT: ...Then, the side length of the octagon is $x\sqrt{2} = \frac{10}{\sqrt{2}+2} \cdot \sqrt{2} = \frac{10\sqrt{2}}{\sqrt{2}+2}$, which simplifies to $10\sqrt{2} - 10 \approx 4.142$ inches. So, 4.14 inches.

Wait, but let me compute $10\sqrt{2} - 10$ more accurately. $\sqrt{2} \approx 1.41421356$, so $10\sqrt{2} \approx 14.1421356$ and $14.1421356 - 10 = 4.1421356$ inches, which rounds to 4.14 inches.
So, the length of one side of the octagon is approximately 4.14.

Figure 13: An example of unfaithful self-verification steps (highlighted in blue) where the TTS score of each step is found smaller than 0.005. Low TTS indicates that those steps are not truly engaged in computation; rather, these reasoning steps are likely to be decorative and function as an appearance of self-verification, contributing minimally to the model’s final prediction.

H EXAMPLES WITH LOW TTS

We provide more examples where reasoning steps have TTS lower than 0.005. We classify each reasoning example based on behavioral patterns labeled in Venhoff et al. (2025)

Uncertainty-estimation example

Context steps (last two sentences): Another way is to divide the quadrilateral into two triangles and find the area of each triangle, then add them up. Looking at the points, maybe I can split the quadrilateral along the x-axis

Steps with low TTS — Uncertainty-estimation: The points (-1, 0) and (2, 0) are on the x-axis, so the quadrilateral is split into two triangles: one above the x-axis and one below.
Wait, actually, the point (0, 1) is above the x-axis, and (0, -3) is below. So, if I connect (-1, 0) to (0, 1) to (2, 0) to (0, -3) and back to (-1, 0), it’s a quadrilateral that crosses over the x-axis.
Alternatively, maybe I can split it into two triangles: one with vertices (-1, 0), (0, 1), (2, 0) and another with vertices (2, 0), (0, -3), (-1, 0). Wait, but that might not cover the entire quadrilateral.
Alternatively, perhaps split it into triangles with a common diagonal. Let me see.
If I connect (0, 1) to (0, -3), that’s a vertical line along the y-axis. Then, the quadrilateral is split into two parts: one on the left side of the y-axis and one on the right.
But actually, the points are (-1, 0), (0, 1), (2, 0), (0, -3)

Uncertainty-estimation example

Context steps (last two sentences): Looking at the points, maybe I can split the quadrilateral along the x-axis. The points (-1, 0) and (2, 0) are on the x-axis, so the quadrilateral is split into two triangles: one above the x-axis and one below

Steps with low TTS — Uncertainty-estimation:
Wait, actually, the point (0, 1) is above the x-axis, and (0, -3) is below. So, if I connect (-1, 0) to (0, 1) to (2, 0) to (0, -3) and back to (-1, 0), it’s a quadrilateral that crosses over the x-axis.
Alternatively, maybe I can split it into two triangles: one with vertices (-1, 0), (0, 1), (2, 0) and another with vertices (2, 0), (0, -3), (-1, 0). Wait, but that might not cover the entire quadrilateral.
Alternatively, perhaps split it into triangles with a common diagonal. Let me see.
If I connect (0, 1) to (0, -3), that’s a vertical line along the y-axis. Then, the quadrilateral is split into two parts: one on the left side of the y-axis and one on the right.
But actually, the points are (-1, 0), (0, 1), (2, 0), (0, -3)

Backtracking example

Context steps (last two sentences): So, putting it all together, 20% of 50% of 80 is 8. That seems correct, but let me verify it another way to make sure I didn’t make a mistake

Steps with low TTS — Backtracking:
Another approach is to multiply all the percentages together first and then apply them to 80. So, 20% is 0. 2, and 50% is 0. 5. Multiplying those together: $0. 2 * 0. 5 = 0. 1$

1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079

Adding-knowledge example

Context steps (last two sentences): Wait, but that might not be accurate. Alternatively, maybe split the quadrilateral into two triangles by drawing a diagonal from (-1, 0) to (2, 0)

Steps with low TTS — Adding-knowledge: Then, the quadrilateral is split into two triangles: one with vertices (-1, 0), (0, 1), (2, 0) and another with vertices (-1, 0), (2, 0), (0, -3). Let me calculate the area of each triangle and add them.

First triangle: (-1, 0), (0, 1), (2, 0)

Using the formula for the area of a triangle with coordinates: $\frac{1}{2} |x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)|$

Plugging in the points:

$$x_1 = -1, y_1 = 0$$

$$x_2 = 0, y_2 = 1$$

$$x_3 = 2, y_3 = 0$$

So,

$$\text{Area} = \frac{1}{2} |(-1)(1 - 0) + 0(0 - 0) + 2(0 - 1)|$$

$$= \frac{1}{2} |(-1)(1) + 0 + 2(-1)|$$

$$= \frac{1}{2} |-1 + 0 - 2|$$

$$= \frac{1}{2} |-3| = \frac{1}{2} * 3 = 1$$

Adding-knowledge example

Context steps (last two sentences): Let me calculate the area of each triangle and add them. First triangle: (-1, 0), (0, 1), (2, 0)

Using the formula for the area of a triangle with coordinates: $\frac{1}{2} |x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)|$

Plugging in the points:

$$x_1 = -1, y_1 = 0$$

$$x_2 = 0, y_2 = 1$$

$$x_3 = 2, y_3 = 0$$

So,

$$\text{Area} = \frac{1}{2} |(-1)(1 - 0) + 0(0 - 0) + 2(0 - 1)|$$

$$= \frac{1}{2} |(-1)(1) + 0 + 2(-1)|$$

$$= \frac{1}{2} |-1 + 0 - 2|$$

$$= \frac{1}{2} |-3| = \frac{1}{2} * 3 = 1.5$$

Steps with low TTS — Adding-knowledge: Second triangle: (-1, 0), (2, 0), (0, -3)

Again, using the same formula:

$$x_1 = -1, y_1 = 0$$

$$x_2 = 2, y_2 = 0$$

$$x_3 = 0, y_3 = -3$$

$$\text{Area} = \frac{1}{2} |(-1)(0 - (-3)) + 2((-3) - 0) + 0(0 - 0)|$$

$$= \frac{1}{2} |(-1)(3) + 2(-3) + 0|$$

$$= \frac{1}{2} |-3 - 6 + 0|$$

$$= \frac{1}{2} |-9| = \frac{1}{2} * 9 = 4.5$$

Adding both areas: $1.5 + 4.5 = 6$

Okay, so that's the same result as before