HIDDEN DOUBT, MISSED CORRECTION: A CAUSAL PATHWAY IN CHAIN-OF-THOUGHT REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

The errors made by large language models (LLMs) in chain-of-thought (CoT) reasoning are often decodable by linear probes from their activations. Yet the error signals in the representations do not always cause the model to self-correct. We study the gap between *error awareness* and *error correction* in LLMs. The Jacobian lens (J-lens) is an interpretability tool that maps activations to tokens the model is poised to verbalize. The mapping uses an averaged Jacobian from the activation to the final layer, which aggregates downstream paths through the model's computation graph across layers and subsequent token positions. Using the J-lens, we find that erroneous CoT steps exhibit stronger representations of doubt-related tokens, such as `But` and `However`. Yet these tokens receive little probability mass in the next-token distribution. To verify whether the hidden doubt causally triggers self-correction, we steer activations along the corresponding J-lens directions, which makes the model voice the doubt and revise the reasoning path, improving final-answer accuracy. We also steer along the probe direction to amplify the error signals and find that this amplification can support self-correction by eliciting doubt. The two intervention approaches support hidden doubt as a possible mechanistic link between error awareness and self-correction. We further compare against non-doubt token interventions and direct text insertion, supporting that the effect is specific to eliciting doubt.

1 INTRODUCTION

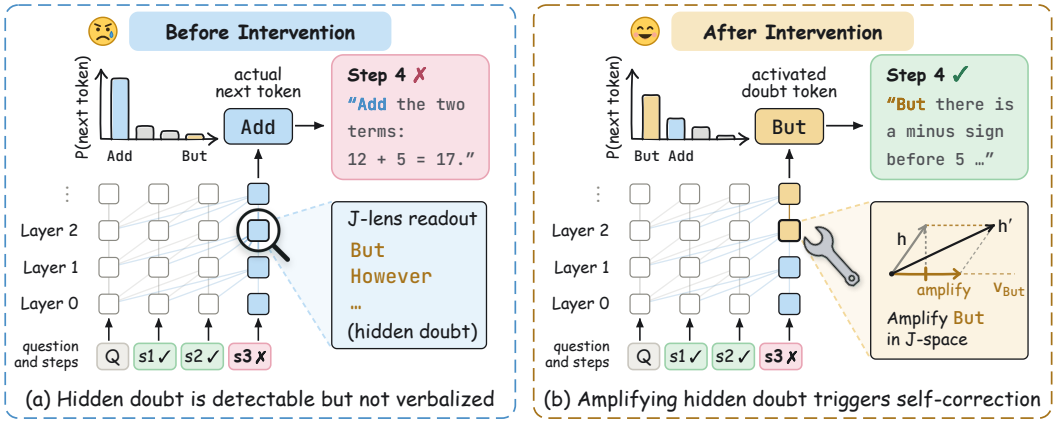


Figure 1: Overview of the hidden-doubt pathway from error awareness to self-correction. (a) After an erroneous CoT step, the J-lens reads out doubt-related tokens such as `But` and `However` from the activation, but the model assigns them low next-token probability and continues the erroneous reasoning. (b) Amplifying the J-lens coordinate of the doubt token `But` increases its next-token probability, leading the model to voice the doubt and self-correct.

Chain-of-thought (CoT) reasoning enables large language models (LLMs) to solve complex problems by generating intermediate reasoning steps (Wei et al., 2022; Kojima et al., 2022). However,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

errors may arise in intermediate steps and propagate through subsequent reasoning. Self-correction can improve reasoning performance and reliability by recognizing and repairing errors in the model's own reasoning (Madaan et al., 2023; Shinn et al., 2023). Existing self-correction methods usually rely on explicit self-evaluation or external feedback, while intrinsic self-correction without reliable feedback remains challenging (Huang et al., 2024; Kamoi et al., 2024).

Growing evidence shows that LLMs internally encode information about the correctness of their own reasoning. In particular, probes are commonly used to assess whether a specific property can be decoded from model activations (Alain & Bengio, 2016; Belinkov, 2022). They are used to detect internal representations regarding correctness and truthfulness (Azaria & Mitchell, 2023; Marks & Tegmark, 2024; Kumaran et al., 2026), as well as hallucination (Orgad et al., 2025; Han et al., 2025). For CoT reasoning, probes can effectively verify trace-level or step-level correctness (Zhang et al., 2025; Sun et al., 2025; Yuan et al., 2026; Zhao et al., 2026; Liang et al., 2026).

Despite the existing evidence on error awareness of LLMs, it remains unclear whether the decodable error signals in CoT reasoning can causally contribute to self-correction. Yuan et al. (2026) argue that these signals are diagnostic rather than causal because their interventions fail to produce correction. Although Sun et al. (2025) use a probe to identify erroneous reasoning steps and then prompt the model to regenerate them, the correction is triggered externally without showing the causal connection between the decoded signal and error correction. In other words, existing approaches either reveal error awareness without turning it into correction or obtain correction through external control. To develop a mechanistic account of this gap, we ask two questions: **(Q1)** *How does the model internally express its awareness of an error?* **(Q2)** *If we can identify such an expression, can we use it to improve the model's ability to self-correct?*

To answer **Q1**, we use the Jacobian lens (J-lens) (Gurnee et al., 2026) to examine how error awareness appears in the residual stream activations, as illustrated in Figure 1. Using the J-lens, we find that erroneous CoT steps carry stronger representations of doubt-related tokens such as `But` and `However`. This doubt, however, rarely reaches generation, as the model assigns these tokens little next-token probability and continues along the wrong reasoning path. To answer **Q2**, we amplify the J-lens coordinates of these doubt-related tokens in the activations. We show that it makes the model voice the doubt and revise the reasoning path. Across three LLMs and three reasoning datasets, the intervention consistently improves recovery on reasoning traces containing an erroneous step. We also amplify the error-related directions identified by probes and observe the same pattern, with doubt surfacing and correction becoming more likely. Our main contributions are:

- We identify doubt-related tokens as a vocabulary-grounded expression of error awareness.
- We show that amplifying hidden doubt triggers its verbalization and improves recovery from erroneous steps, revealing one way error awareness can lead to self-correction.
- More broadly, we demonstrate a way to apply the J-lens to obtain token-level interpretations of internal representations and verify them through causal interventions, complementing probes by grounding decodable properties in vocabulary tokens.

2 Motivation: Step-Level CoT Errors Are Decodable

2.1 Experimental Setup and Probes

Step-Level CoT Verification. Given a problem and an LLM-generated CoT trace through the current step, *step-level CoT verification* aims to determine whether the current step contains an error. A step is labeled as erroneous if it introduces a mistake, such as an arithmetic error, the use of an incorrect operand, and an invalid reasoning operation.

Data. We use the three datasets from Zhao et al. (2026), including two synthetic datasets for arithmetic and boolean reasoning (Arithmetic and Boolean), and the test split of GSM8K, a dataset of grade school mathematics word problems (Cobbe et al., 2021). We evaluate CoTs generated by Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, and Gemma-2-9B-IT. For Llama-3.1-8B-Instruct, we use the CoTs and step-level annotations released with these datasets. For the other models, we generate CoTs on the same problems and annotate each step using the same procedure. Following Zhao et al. (2026), the annotations for the synthetic datasets combine judgments from Llama-3.3-70B-Instruct and a programmatic verifier, whereas those for GSM8K rely only on the LLM judge. For

traces containing errors, we retain the reasoning through the first erroneous step and discard all subsequent steps; correct traces are kept in full. We split the traces into 80% training and 20% test sets, keeping all steps from the same trace in the same split. Dataset statistics are provided in Table 5 in the Appendix. All experiments in the paper were conducted on NVIDIA H200 GPUs.

Linear Probes. Linear probes are simple linear classifiers used to assess whether a particular property can be decoded from a model's internal representations (Alain & Bengio, 2016; Belinkov, 2022). In our setting, given the residual stream activation $\boldsymbol{h}_\ell$ at layer ℓ, taken at the final token of the step, we train a logistic regression classifier to predict the step-level correctness label:

$$P(y = 1 \mid \boldsymbol{h}_\ell) = \sigma(\boldsymbol{w}_\ell^\top \boldsymbol{h}_\ell + b_\ell), \tag{1}$$

where $y = 1$ indicates that the current CoT step is erroneous, σ is the logistic sigmoid function, and $\boldsymbol{w}_\ell$ and b_ℓ are the learned parameters.

2.2 Verification Results

Figure 2 shows the performance of probes for step-level CoT verification. The probes achieve strong verification performance across models and datasets, with AUROC near or above 0.9 on Arithmetic and Boolean and above 0.8 on GSM8K. These results show that *the correctness of CoT steps is decodable from activations*, consistent with prior trace-level findings (Yuan et al., 2026). It indicates the activations contain information that distinguishes erroneous steps from correct ones. In this limited sense, the model may be *aware* of an error even when it continues along an erroneous reasoning path. This raises a natural question: *How does the model internally express its recognition of an error?* Further, *if we can identify such an expression, can we use it to improve the model's ability to self-correct?*

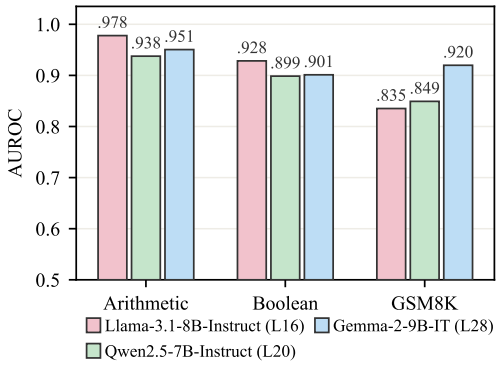


Figure 2: Step-level CoT verification results of linear probes across models and datasets.

3 Hidden Doubt in Erroneous Chain-of-Thought Steps

To answer the first question, we use the J-lens to map residual stream activations to token-level readouts and examine whether there exist token-level signals that distinguish erroneous steps from correct ones. We find that erroneous CoT steps exhibit stronger representations of doubt-related tokens in the J-space. Yet these tokens receive little probability mass even if they rank high in the next-token distribution. This discrepancy suggests the doubt-related signals may be present internally but are not strong enough to influence next-token predictions to trigger self-correction.

3.1 Preliminaries

J-lens. The *J-lens* maps an intermediate residual stream activation to readouts associated with the model's vocabulary, revealing the tokens the model is poised to verbalize (Gurnee et al., 2026). To construct the J-lens for each layer ℓ, a Jacobian $\boldsymbol{J}_\ell$ is estimated to capture the effect of changes in the activation $\boldsymbol{h}_{\ell,t}$ on final-layer activations $\boldsymbol{h}_{L,t'}$ at the current and subsequent positions:

$$\boldsymbol{J}_\ell = \mathbb{E}_{t,\, t' \geq t,\, \text{prompt}} \left[\frac{\partial \boldsymbol{h}_{L,t'}}{\partial \boldsymbol{h}_{\ell,t}} \right]. \tag{2}$$

The expectation is taken over source positions t, subsequent positions $t' \geq t$, and prompts sampled from a pretraining-like distribution. Given an activation $\boldsymbol{h}_{\ell,t}$, the J-lens maps it through this averaged Jacobian $\boldsymbol{J}_\ell$ and decodes it with the model's normalization and unembedding matrix $\boldsymbol{W}_U$:

$$\text{lens}(\boldsymbol{h}_{\ell,t}) = \text{softmax}\left(\boldsymbol{W}_U \,\text{norm}(\boldsymbol{J}_\ell \boldsymbol{h}_{\ell,t})\right). \tag{3}$$

Table 1: Top-4 tokens with the largest differences in J-space selection rate, Δr_u, between erroneous and correct steps, at L18 (Llama), L27 (Qwen), and L26 (Gemma). Shading is proportional to Δr_u, red marks doubt-related tokens, and ␣ marks a leading space.

Llama-3.1-8B-Instruct			Qwen2.5-7B-Instruct			Gemma-2-9B-IT		
Arith.	Bool.	GSM8K	Arith.	Bool.	GSM8K	Arith.	Bool.	GSM8K
␣BUT	␣BUT	␣HOWEVER	␣as	␣Essentially	␣However	␣finally	␣finally	␣therefore
␣NEG	␣Applies	␣BUT	␣is	␣Then	␣or	␣further	␣again	␣unfortunately
␣Therefore	␣concatenated	答案	␣So	␣Evalu	So	␣then	␣resulting	␣because
final	␣Rounded	␣subtract	end	␣But	␣Initially	␣followed	␣overall	␣Alternatively

J-space. At layer ℓ, the *J-lens vector* $\boldsymbol{v}_{\ell,u}$ for token u is the unit-normalized u-th row of $\widetilde{\boldsymbol{W}}_U \boldsymbol{J}_\ell$, where $\widetilde{\boldsymbol{W}}_U$ is obtained by scaling each column of $\boldsymbol{W}_U$ by the corresponding learned scale from the model's final normalization layer. For a sparsity level k, the *J-space* at layer ℓ is the set of nonnegative combinations of at most k J-lens vectors. Following Gurnee et al. (2026), we set $k = 25$ and use gradient pursuit to perform a sparse nonnegative decomposition of each activation into an approximate J-space component $\boldsymbol{h}_J$ and a non-J-space remainder $\boldsymbol{h}_{\text{rem}}$:

$$\boldsymbol{h}_{\ell,t} = \boldsymbol{h}_J + \boldsymbol{h}_{\text{rem}}, \qquad \boldsymbol{h}_J = \sum_{u \in S} a_u \boldsymbol{v}_{\ell,u}, \quad a_u \geq 0, \quad |S| \leq k. \tag{4}$$

3.2 Doubt-Related Tokens Are More Prevalent on Erroneous Steps

For each model and dataset, we analyze all erroneous steps together with an equal number of correct steps matched by step index, i.e., each erroneous step is paired with a correct step at the same position in a CoT trace for a different problem. The numbers of steps are provided in Table 6 in Appendix A. We use the J-lens to read out what the model is poised to verbalize at selected positions and layers. We use pre-fitted J-lenses released by Neuronpedia[1] for Llama and Gemma, and fit the Qwen lens on WikiText-103 with the official J-lens library following the same procedure (Gurnee et al., 2026). To identify token-level signals that distinguish erroneous from correct steps, we compare the selection rates by the J-space sparse decomposition in erroneous steps and correct steps:

$$\Delta r_u = r_u^{\text{erroneous}} - r_u^{\text{correct}}, \tag{5}$$

where $r_u^{\text{erroneous}}$ and r_u^{correct} are the fractions of erroneous and correct steps whose decomposition contains token u, respectively. A positive Δr_u indicates that token u is more prevalent on erroneous steps. Table 1 reports the four tokens with the largest Δr_u for each model and dataset. For each model, we select a single layer to report the selection rates across datasets. For Llama, doubt-related tokens such as ␣BUT and ␣HOWEVER have the largest Δr_u across all three datasets. This pattern is less dominant for Qwen and Gemma. ␣But and ␣However appear among the most enriched tokens for Qwen, while ␣unfortunately appears for Gemma, but only on some datasets. This qualitative observation motivates our systematic analysis of doubt-related tokens below.

To determine whether doubt-related signals are more prevalent on erroneous than correct steps, we use a fixed lexicon of doubt-related tokens (see Table 7 in the Appendix) and compute the fraction of steps whose J-space decomposition selects at least one token from the lexicon, as shown in Figure 3. Across all three models and datasets, erroneous steps are more likely to select doubt-related tokens in their J-space decomposition, with the largest differences observed for Llama. Although the selection rates for erroneous steps are lower for Qwen and Gemma than for Llama, the corresponding rates for correct steps remain below 10% across all datasets, leaving a clear gap between erroneous and correct steps. This consistent gap of the selection rates between erroneous and correct steps indicates that *doubt-related tokens are a way models internally express their awareness of an error*.

To trace doubt-related signals across model depth, we measure their selection rates r_u at every layer. Figure 4 shows the layerwise selection rates on GSM8K for both the full doubt-related lexicon and the individual dominant token for each model. We define each model's dominant doubt token as the one with the largest peak J-space selection-rate gap averaged across datasets, yielding ␣BUT for

[1] https://huggingface.co/neuronpedia/jacobian-lens

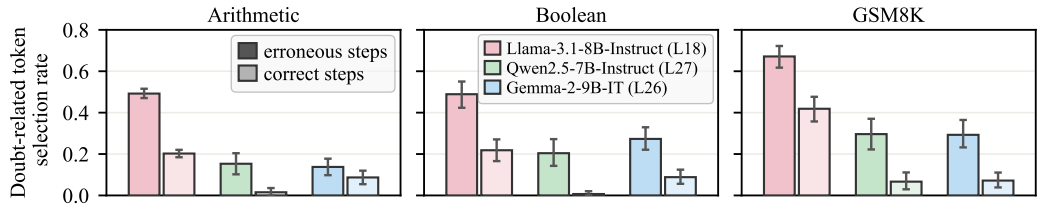


Figure 3: The selection rates of doubt-related tokens in the J-space decomposition for erroneous and correct steps at each model's selected layer. Error bars indicate 95% confidence intervals.

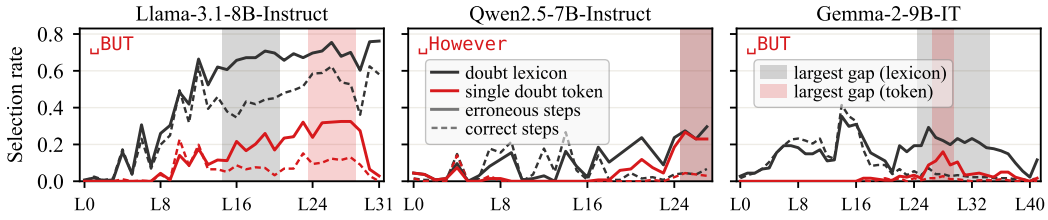


Figure 4: Layerwise fraction of steps whose J-space decomposition selects any token from the doubt lexicon (black) or the dominant doubt-related token for each model (red) on GSM8K. For each curve pair, shaded bands mark the contiguous layers around the peak.

Llama and Gemma and ␣However for Qwen. Corresponding results for Arithmetic and Boolean are provided in Figure 9 in the Appendix. In lower layers, doubt-related token selection rates show little separation between erroneous and correct steps, with Qwen occasionally exhibiting higher rates for correct steps. In the middle and higher layers, the two groups become clearly separated across all three models, with consistently higher rates for erroneous steps. Across models, Llama shows relatively high selection rates for both erroneous and correct steps, whereas in Qwen and Gemma, the selection rates for correct steps approach zero in the middle and higher layers. The layer at which the gap between erroneous and correct steps peaks is model-specific. Specifically, the gap peaks in middle and higher layers for Llama and Gemma, but closer to the final layer for Qwen.

3.3 Why Are Doubt-Related Tokens Not Verbalized?

As shown above, erroneous steps carry clear signals aligned with doubt tokens. Yet the models rarely verbalize these tokens and instead continue reasoning along the erroneous path, as reported in Appendix C.3. Thus, we compute the next-token distribution after each erroneous step. Table 2 reports the median values of the rank and probability of the highest-ranked doubt-related token across steps. These tokens typically rank highly, especially for Llama, but their probabilities are consistently small compared with those of the top-ranked tokens. This suggests that internal doubt reaches the output distribution but is not strong enough to trigger self-correction, which motivates our interventions.

Table 2: Median rank of the highest-ranked doubt-related token after erroneous steps and the order of magnitude of its median probability P_{doubt}, alongside the median maximum next-token probability $P_{\max}$.

Model	Arithmetic			Boolean			GSM8K		
	rank	P_{doubt}	$P_{\max}$	rank	P_{doubt}	$P_{\max}$	rank	P_{doubt}	$P_{\max}$
Llama	6	10^{-4}	0.73	6	10^{-4}	0.98	3	10^{-4}	0.93
Qwen	54	10^{-7}	0.76	66	10^{-9}	1.00	29	10^{-7}	0.86
Gemma	113	10^{-9}	1.00	72	10^{-8}	1.00	67	10^{-9}	1.00

4 From Hidden Doubt to Self-Correction

We now investigate whether amplifying the hidden doubt in LLMs can trigger self-correction. First, we intervene along J-lens vectors to amplify doubt, which makes the models verbalize doubt and improves their recovery from erroneous steps. Second, we amplify error signals identified by linear probes and find that they can support self-correction by eliciting the doubt-related tokens. Third, we

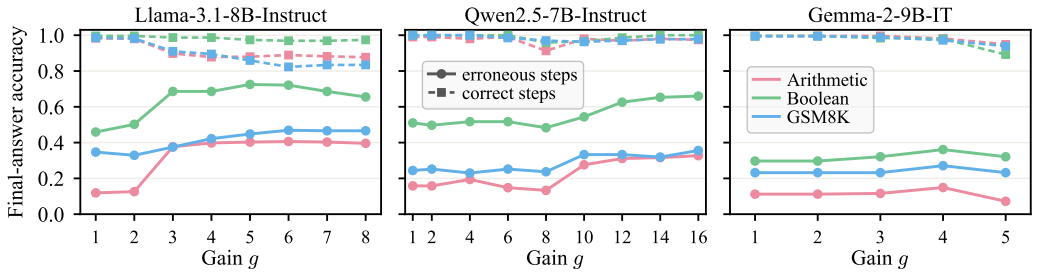


Figure 5: Final-answer accuracy when intervening after erroneous steps (solid) and correct steps (dashed) by scaling the J-lens coordinate of the doubt token (␣BUT for Llama, ␣However for Qwen, and ␣But for Gemma) with gain g. $g = 1$ denotes greedy continuation without intervention.

compare against interventions along non-doubt tokens to show the specificity of doubt tokens for self-correction. Finally, we compare J-lens interventions with inserting doubt-related text.

4.1 Intervention Setup

J-lens Intervention. Given an activation $\boldsymbol{h}_{\ell,t}$ at position t in layer ℓ and a target token u, we scale the token's positive J-lens coordinate $c_{\ell,t,u}$ by a gain $g \geq 1$:

$$c_{\ell,t,u} = \boldsymbol{v}_{\ell,u}^{\top} \boldsymbol{h}_{\ell,t}, \qquad \boldsymbol{h}'_{\ell,t} = \boldsymbol{h}_{\ell,t} + (g-1)\max\big(c_{\ell,t,u}, 0\big)\boldsymbol{v}_{\ell,u}. \tag{6}$$

Here, $\boldsymbol{v}_{\ell,u}$ is the unit J-lens vector and $c_{\ell,t,u}$ is its J-lens coordinate. Our intervention leaves the activation unchanged when the coordinate is nonpositive.

Data. We evaluate all erroneous steps in the datasets with an equal number of correct steps matched by step index. Unlike the data used in Section 3, we draw correct steps from traces with correct final answers to reduce the effect of calculation errors on final-answer accuracy, and exclude final steps because our evaluation requires the model to generate a continuation after the selected step. Step counts are provided in Table 6 in the Appendix.

4.2 Amplifying Doubt Triggers Self-Correction

For each step to intervene, the model input consists of the problem and the CoT up to and including that step, followed by a newline \n\n. We apply the intervention in Eq. (6) at position t of this newline. The model then greedily generates the continuation after the newline for up to 1,024 tokens. Continuations without an answer are treated as incorrect. We apply the intervention at three middle-to-late layers chosen according to the region where Figure 4 shows the largest differences between erroneous and correct steps in doubt-related token selection rates. Specifically, we use L18, L21, and L26 for Llama, L24–L26 for Qwen, and L26, L29, and L32 for Gemma. For the intervention, we use representative doubt markers, namely ␣BUT for Llama, ␣However for Qwen, and ␣But for Gemma.

Figure 5 shows final-answer accuracy of continuations following erroneous and correct steps across intervention gains. For erroneous steps, the accuracy generally rises with g after a model-specific threshold, but declines when the amplification becomes too strong. For correct steps, the accuracy decreases modestly overall, with a more pronounced decline for Llama. Qwen shows a marked drop at $g = 8$ but then recovers and remains stable. The results show that at an appropriate gain, the intervention improves recovery from erroneous steps while causing only limited degradation after correct steps. Across models, Llama shows the largest changes, Gemma is the most stable, and Qwen shows the strongest selectivity, with clear recovery from erroneous steps and little degradation after correct ones. The recovery effects under intervention broadly track the strength of doubt-related representations in Figure 4.

To further analyze the effects of our J-lens intervention, we list verbalization rates of the doubt token and final-answer accuracy after intervened erroneous and correct steps in Table 3. For each model, we use a single gain $g^{\star}$ across all datasets, chosen from the sweep in Figure 5 based on the

Table 3: J-lens intervention results for continuations following erroneous and correct steps, using $g^\star = 5$, 16, and 4 for Llama, Qwen, and Gemma, respectively. Verbalization rate is the fraction of continuations beginning with the amplified token. Before and After denote final-answer accuracy at $g = 1$ and $g^\star$. Green and red indicate accuracy increases and decreases.

Model	Dataset	Verbalization rate		Acc. (Err. steps)			Acc. (Corr. steps)		
		Err.	Corr.	Before	After	Δ	Before	After	Δ
Llama-3.1-8B-Instruct	Arithmetic	1.00	1.00	0.12	0.40	+0.28	0.98	0.88	−0.10
	Boolean	1.00	1.00	0.46	0.72	+0.27	1.00	0.97	−0.02
	GSM8K	0.87	0.70	0.35	0.45	+0.10	0.99	0.86	−0.13
Qwen2.5-7B-Instruct	Arithmetic	1.00	1.00	0.16	0.33	+0.17	0.99	0.97	−0.02
	Boolean	1.00	1.00	0.51	0.66	+0.15	1.00	1.00	0.00
	GSM8K	1.00	0.99	0.24	0.36	+0.11	1.00	0.98	−0.02
Gemma-2-9B-IT	Arithmetic	0.39	0.15	0.11	0.15	+0.04	1.00	0.98	−0.01
	Boolean	0.58	0.41	0.30	0.36	+0.06	1.00	0.98	−0.02
	GSM8K	0.52	0.14	0.23	0.27	+0.04	0.99	0.97	−0.02

aggregate trade-off between recovery after erroneous steps and degradation after correct steps. This gives $g^\star = 5$, 16, and 4 for Llama, Qwen, and Gemma. The intervention elicits doubt tokens in most continuations for Llama and Qwen and less frequently for Gemma, indicating that the amplified internal representation is effectively expressed in the generated continuation. Across all models and datasets, it improves accuracy when applied after erroneous steps while causing only limited degradation when applied after correct steps. Llama shows the largest gains on erroneous steps but also the largest cost on correct steps. Qwen is the most selective, with gains of 0.11 to 0.17 at a cost of at most 0.02. Gemma changes the least, consistent with the weaker doubt signal in its J-space decompositions. These results show that *amplifying the internal representation of doubt can trigger self-correction*.

4.3 Amplifying Error Representations of Probes Elicits Doubt

Recalling the results in Section 2, the trained linear probes effectively distinguish erroneous from correct steps, indicating the probes can extract error-related information. Given that our J-lens intervention *amplifies* the component of an activation along a doubt-related direction, we investigate whether similarly amplifying the error-related direction identified by the probes can support self-correction. To test this, we intervene along the normalized probe direction $\hat{\boldsymbol{w}}_\ell = \boldsymbol{w}_\ell / \|\boldsymbol{w}_\ell\|_2$. Intuitively, since erroneous steps are labeled $y = 1$, amplifying along $\hat{\boldsymbol{w}}_\ell$ strengthens the error signals in activations. Because Eq. (6) amplifies only positive coordinates, we center the probe projection. Specifically, we set the zero point μ_ℓ to the midpoint between the mean probe projections of erroneous and correct steps at the step boundary on the training split. The equation for the probe intervention is

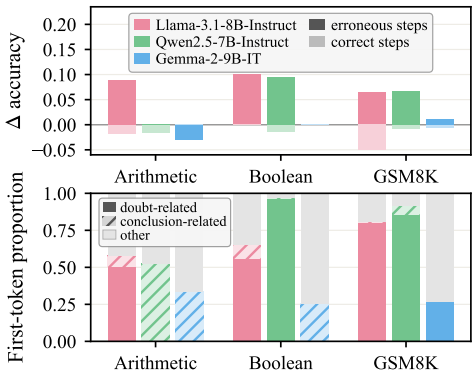


Figure 6: Probe intervention. **Top**: changes in accuracy after intervening. **Bottom**: first-token proportions after erroneous steps with $c_{\ell,t} > 0$, categorized as doubt-related, conclusion-related, and other.

$$c_{\ell,t} = \hat{\boldsymbol{w}}_\ell^\top \boldsymbol{h}_{\ell,t} - \mu_\ell, \qquad \boldsymbol{h}'_{\ell,t} = \boldsymbol{h}_{\ell,t} + (g-1)\max\big(c_{\ell,t}, 0\big)\hat{\boldsymbol{w}}_\ell. \tag{7}$$

The probe intervention uses the same positions, layers, and gains as the J-lens intervention. We measure changes in final-answer accuracy relative to the un-intervened baseline after erroneous and matched correct steps. The top panel of Figure 6 shows that amplifying the probe direction improves recovery from erroneous steps across all datasets for Llama and on Boolean and GSM8K for Qwen, while the effects on Gemma are small. The accuracy after correct steps remains unchanged or

decreases slightly. This suggests that probe-identified error directions can support self-correction, with effects varying across settings.

To examine the model's immediate response to the intervention, we categorize the first generated token after erroneous steps for which $c_{\ell,t} > 0$ (i.e., the probe direction is amplified), as doubt-related, conclusion-related, and other. The bottom panel of Figure 6 shows that doubt tokens account for a large share of first tokens in some cases, especially for Qwen on Boolean and GSM8K. Taken together, the two panels show a strong correspondence between doubt verbalization and recovery after erroneous steps; Table 9 in the Appendix confirms that recovery is concentrated in continuations that verbalize doubt. That is, *amplifying the error-related directions identified by the probes can promote self-correction by eliciting doubt.* Despite being learned without token-level supervision, the probe directions elicit the same doubt implicated by the J-lens readout.

Interestingly, Yuan et al. (2026) argue that probes can detect reasoning errors, yet their interventions do not improve self-correction. Their intervention subtracts the error-related component from the activation as $\boldsymbol{h}'_{\ell,t} = \boldsymbol{h}_{\ell,t} - \alpha(\hat{\boldsymbol{w}}_\ell^\top \boldsymbol{h}_{\ell,t})\hat{\boldsymbol{w}}_\ell$, where $\alpha > 0$. Conceptually, this is the suppressive counterpart of our intervention, corresponding to $g < 1$. Subtracting a direction associated with error may seem like the natural way to correct the model, but our results point to the opposite intervention. Rather than erasing the model's error awareness, amplifying it can bring that awareness to the surface as doubt and allow the model to self-correct.

4.4 Is Self-Correction Specific to Doubt Tokens?

We test whether self-correction is specific to doubt tokens or can also be induced by amplifying other token directions. We use the same J-space readout results as in Section 3. We remove noisy tokens that do not appear in the model's own CoTs, then select, for each model, the five remaining non-doubt lexical tokens with the largest average Δr_u across datasets. We amplify each token using the same layers, position, and gain as the doubt token and evaluate it on the same erroneous and correct steps. Figure 7 shows accuracy changes for Llama averaged across the three datasets. The non-doubt interventions yield little recovery from erroneous steps, whereas the ␣BUT intervention clearly improves recovery. Most also reduce accuracy when applied after correct steps, with some even causing greater degradation than the doubt intervention. These results suggest that the recovery benefit is specific to doubt-related directions, rather than a general effect of amplifying token directions that are more prominent on erroneous steps. Full results are provided in Table 10 in the Appendix.

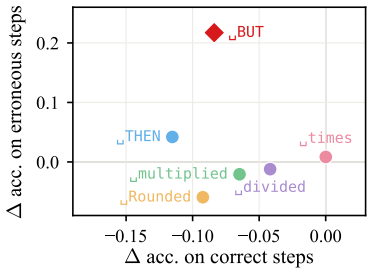


Figure 7: Effect of amplifying non-doubt tokens in J-space for Llama.

4.5 Amplifying Doubt Matches Oracle Text Insertion

Having shown that amplifying a doubt-related J-lens coordinate can trigger self-correction, we ask whether it can achieve the recovery obtained by text insertion. We compare the J-lens intervention with a text-level oracle that inserts a doubt-related token at the same step boundary. As controls, we insert the double-check instruction "Let me double-check the previous step." to test the effect of generic reconsideration, and resample at temperature 0.7 to test whether any improvement arises merely from generating a different continuation. Table 4 reports the changes in final-answer accuracy for Llama, with full results for Qwen and Gemma provided in Table 11 in the Appendix. Directly inserting doubt produces

Table 4: Comparison of J-lens intervention and text insertion for Llama. Values show changes in accuracy relative to greedy continuation after erroneous and correct steps.

Method	Arithmetic		Boolean		GSM8K	
	Err.	Corr.	Err.	Corr.	Err.	Corr.
J-lens ␣BUT	+0.28	−0.10	+0.27	−0.02	+0.10	−0.13
Insert ␣BUT	+0.25	−0.11	+0.25	0.00	+0.12	−0.16
Insert ␣However	+0.33	−0.15	+0.27	−0.03	+0.09	−0.19
Insert ␣Wait	+0.13	−0.08	+0.18	−0.03	+0.14	−0.23
Insert double-check	+0.11	−0.02	+0.16	−0.03	+0.09	−0.05
Resample (T=0.7)	+0.01	−0.01	−0.01	−0.01	0.00	0.00

accuracy gains comparable to the J-lens intervention, whereas the double-check instruction is less effective and resampling brings little improvement. Among the doubt-related tokens, ␣BUT and ␣However yield larger gains, while ␣Wait, which has been used to elicit further reasoning and self-correction (Muennighoff et al., 2025; Tsui, 2025), is less effective. Thus, amplifying doubt already present in the model can promote recovery as effectively as explicitly supplying doubt. Unlike text insertion, the J-lens intervention amplifies a doubt-related direction in the activation, bringing the model's latent doubt into generation without forcing a token into the context.

5 Related Work

Error Awareness and Probing. Probes can detect signals related to correctness and truthfulness (Burns et al., 2023; Azaria & Mitchell, 2023; Marks & Tegmark, 2024; Kumaran et al., 2026), as well as hallucination (Orgad et al., 2025). For CoT reasoning, probes can predict trace- or step-level correctness (Zhang et al., 2025; Sun et al., 2025; Yuan et al., 2026; Liang et al., 2026), while Zhao et al. (2026) decode step correctness from features of the attribution graph. The classification results can be used to stop reasoning after reaching a correct intermediate answer (Zhang et al., 2025), regenerate detected erroneous steps (Sun et al., 2025), and prune unpromising traces (Liang et al., 2026). Our work starts from the same decodability but asks what the error signal is in token space and why it fails to affect generation.

Self-Correction in CoT Reasoning. Prompting methods such as Self-Refine (Madaan et al., 2023) and Reflexion (Shinn et al., 2023) ask the model to critique and revise its own output, but intrinsic self-correction without external feedback is often ineffective for reasoning (Huang et al., 2024; Kamoi et al., 2024). Later work induces the ability mainly through training (Kumar et al., 2025; Guo et al., 2025) and by forcing reflection tokens such as Wait into the text (Muennighoff et al., 2025). However, these methods realize self-correction at the behavior level. In contrast, we provide a mechanistic account of self-correction by tracing it to the model's internal error awareness. Intervening on latent doubt achieves improvements comparable to directly inserting a doubt token, and But and However are more effective than Wait in our setting.

Activation Steering. Activation steering changes model behavior at inference time by modifying residual stream activations along directions derived from probes (Li et al., 2023), contrastive prompt pairs (Turner et al., 2023; Rimsky et al., 2024), or representation reading (Zou et al., 2023). Its effects can vary across inputs and concepts (Tan et al., 2024). Recent work has extended steering to CoT reasoning, for example by eliciting backtracking in reasoning models (Venhoff et al., 2025). Sun et al. (2025) use probe predictions to locate erroneous arithmetic steps and prompt the model to regenerate from those steps, but the correction is externally initiated rather than caused by an intervention on the decoded signal. Yuan et al. (2026) suppress activation components along probe directions associated with CoT errors but do not induce self-correction, leading them to characterize decodable error awareness as diagnostic rather than causal. In fact, we find that amplifying the activation component along the same direction, rather than suppressing it, triggers self-correction.

6 Conclusion

In this paper, we study the gap between error awareness and self-correction of LLMs in CoT reasoning by interpreting and intervening on model activations with the J-lens and probes. We first show that doubt-related tokens are more prevalent in the J-space decompositions of erroneous than correct CoT steps, but these tokens receive little next-token probability and usually remain unvoiced. We then intervene by amplifying the corresponding J-lens coordinates, causing the model to voice its doubt and reconsider its preceding reasoning and yielding higher final-answer accuracy. These results support latent doubt as one possible mechanism linking error awareness to self-correction.

Although our focus is mechanistic understanding rather than performance optimization, future work could build on these findings to design more refined J-lens interventions, for example by applying a coordinate threshold or learning a coordinate-to-gain mapping. On the other hand, future work could extend the analysis to reasoning models to determine whether they are better able to verbalize latent doubt without intervention. Evaluations on additional non-synthetic benchmarks beyond GSM8K would also help test the generalizability of our findings.

AI USE STATEMENT

In this work, we used generative AI tools to assist in implementing methods. We have not used generative AI tools to develop theoretical models or conceptual frameworks, to propose or refine hypotheses, to design or provide feedback on research methodology or experiments, or to interpret results. The remaining tasks with required disclosure (generating synthetic datasets, formulating or proving mathematical claims, translation, cleaning and reformatting datasets, and qualitative or thematic data analysis) are not applicable to this work. Additionally, we used generative AI tools to edit the paper to improve readability. We have reviewed all AI-assisted work. All code was inspected by the authors. All AI-edited text was reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the internal representations of language models on mathematical and logical reasoning tasks. All datasets are open-source, namely GSM8K and the synthetic arithmetic and boolean reasoning datasets, and all models are open-weight (Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, and Gemma-2-9B-IT) and were run locally under their respective licenses. No human subjects, personal data, or crowdsourced annotation are involved.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations (ICLR)*, 2023.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, et al. Verbalizable representations form a global workspace in language models. *arXiv preprint arXiv:2607.15495*, 2026.

Jiatong Han, Neil Band, Muhammed Razzak, Jannik Kossen, Tim G. J. Rudner, and Yarin Gal. Simple factuality probes detect hallucinations in long-form natural language generation. In *EMNLP (Findings)*, pp. 16209–16226, 2025.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations (ICLR)*, volume 2024, pp. 32808–32824, 2024.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 22199–22213, 2022.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 54523–54549, 2025.

Dharshan Kumaran, Viorica Patraucean, Simon Osindero, Petar Veličković, and Nathaniel Daw. How LLMs detect and correct their own errors: The role of internal confidence signals. *arXiv preprint arXiv:2604.22271*, 2026.

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

Zhixiang Liang, Beichen Huang, Zheng Wang, and Minjia Zhang. Hidden states as early signals: Step-level trace evaluation and pruning for efficient test-time scaling. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26800–26813, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 46534–46594, 2023.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model true/false datasets. In *Conference on Language Modeling (COLM)*, 2024.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, 2025.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 66880–66913, 2025.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 15504–15522, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 8634–8652, 2023.

Yucheng Sun, Alessandro Stolfo, and Mrinmaya Sachan. Probing for arithmetic errors in language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 8111–8128, 2025.

Daniel Tan, David Chanin, Aengus Lynch, Brooks Paige, Dimitrios Kanoulas, Adrià Garriga-Alonso, and Robert Kirk. Analysing the generalisation and reliability of steering vectors. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 139179–139212, 2024.

Ken Tsui. Self-correction bench: Uncovering and addressing the self-correction blind spot in large language models. *arXiv preprint arXiv:2507.02778*, 2025.

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

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 24824–24837, 2022.

Aojie Yuan, Zhiyuan Julian Su, Haiyue Zhang, Yi Nian, and Yue Zhao. Hidden error awareness in chain-of-thought reasoning: The signal is diagnostic, not causal. *arXiv preprint arXiv:2605.09502*, 2026.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*, 2025.

Zheng Zhao, Yeskendir Koishekenov, Xianjun Yang, Naila Murray, and Nicola Cancedda. Verifying chain-of-thought reasoning via its computational graph. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 122812–122839, 2026.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Dataset Statistics

Table 5 summarizes the traces and steps used in our experiments. For traces containing an error, we retain the reasoning through the first erroneous step, so each such trace contributes one erroneous step; correct traces are retained in full. We split the data 80%/20% at the trace level, keeping all steps from a trace in the same split. The training split is used to fit the linear probes and estimate the centering statistic for probe intervention, while the test split is used to report probe AUROC.

Table 5: Dataset statistics by model, dataset, and split.

Model	Dataset	All				Train		Test	
		Traces	Steps	Err.	Err. (%)	Steps	Err.	Steps	Err.
Llama-3.1-8B-Instruct	Arithmetic	24,501	155,631	1,843	1.18	124,351	1,492	31,280	351
	Boolean	15,297	126,962	229	0.18	101,497	183	25,465	46
	GSM8K	1,319	8,073	277	3.43	6,418	216	1,655	61
Qwen2.5-7B-Instruct	Arithmetic	22,561	154,128	196	0.13	123,246	153	30,882	43
	Boolean	12,407	96,353	147	0.15	77,001	120	19,352	27
	GSM8K	1,319	7,755	135	1.74	6,219	101	1,536	34
Gemma-2-9B-IT	Arithmetic	22,932	117,783	276	0.23	94,210	218	23,573	58
	Boolean	12,997	79,964	249	0.31	63,933	204	16,031	45
	GSM8K	1,319	6,914	181	2.62	5,515	135	1,399	46

Table 6 lists, for each model and dataset, the numbers of erroneous steps and matched correct steps used in the J-space readout of Section 3 and in the intervention experiments of Section 4. The two analyses use the same erroneous steps; the intervention experiments draw the matched correct steps from traces with correct final answers, excluding final steps.

Table 6: Numbers of erroneous and matched correct steps used in the J-space readout and the intervention experiments.

Model	Dataset	Erroneous steps	Correct steps
Llama-3.1-8B-Instruct	Arithmetic	1,843	1,843
	Boolean	229	229
	GSM8K	277	277
Qwen2.5-7B-Instruct	Arithmetic	196	196
	Boolean	147	147
	GSM8K	135	135
Gemma-2-9B-IT	Arithmetic	276	276
	Boolean	249	249
	GSM8K	181	181

B Additional Results on CoT Verification

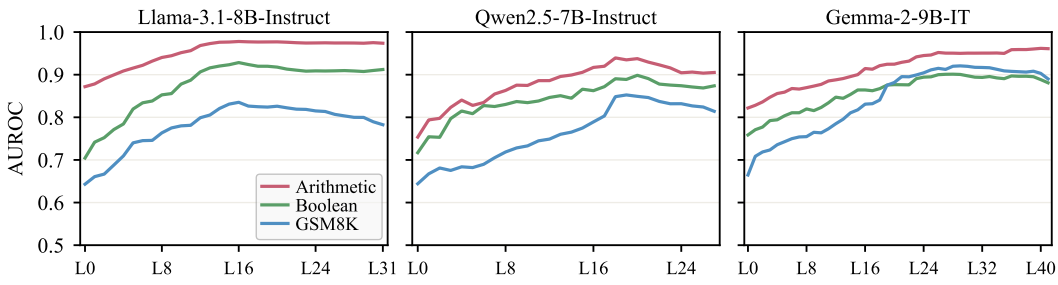


Figure 8: Layerwise linear probe AUROC for step-level CoT verification.

Figure 8 shows that step correctness becomes increasingly decodable from early to middle-to-late layers across all models and datasets. AUROC then generally plateaus or declines slightly near the final layers, suggesting that the most separable error-related representations often emerge before the model's output layer. Arithmetic is consistently the easiest to verify, whereas GSM8K is generally more challenging, especially for Llama and Qwen.

C Additional Details on J-space Readout

C.1 Doubt-Related Token Lexicon

We use the fixed lexicon in Table 7 for the analyses in Table 1 and Figure 3. A vocabulary token is classified as doubt-related when its decoded string matches a listed word after surrounding whitespace is stripped and the string is lowercased. This yields 55 matching tokens for Llama-3.1-8B-Instruct, 55 for Qwen2.5-7B-Instruct, and 68 for Gemma-2-9B-IT.

Table 7: Doubt-related word lexicon.

Normalized word forms					
`but`	`however`	`wait`	`actually`	`hmm`	`oops`
`incorrect`	`wrong`	`mistake`	`mistaken`	`doesn`	`≠`
`corrected`	`correction`	`recheck`	`reconsider`	`instead`	`unfortunately`

C.2 Layerwise Doubt Representations

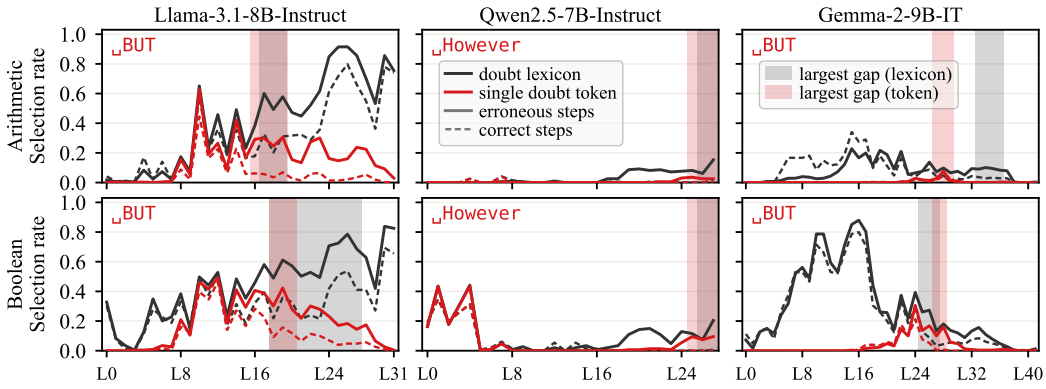


Figure 9: Doubt across depth on Arithmetic (top) and Boolean (bottom).

Figure 9 reports the layerwise J-space selection rates of the doubt lexicon and the dominant doubt-related token for erroneous and correct steps on Arithmetic and Boolean. Llama shows the clearest separation between erroneous and correct steps on both datasets, whereas the strength and location of the gap are more dataset-dependent for Qwen and Gemma. The largest gaps generally occur in middle-to-late layers but at different depths across models, supporting our use of model-specific layers for intervention.

C.3 Doubt Verbalization Without Intervention

Table 8 reports doubt verbalization without intervention. For each erroneous step and its matched correct step, we greedily decode up to 1,024 tokens from the step boundary. We report whether a doubt-related token appears first, a doubt phrase occurs in the next step, or a doubt phrase occurs anywhere in the continuation. For the Next step and Anywhere metrics, we search for the lexicon words in Table 7 together with the additional phrases *isn't*, *re-evaluate*, *double-check*, *should be*, and *let me check/verify*. Across models and datasets, doubt almost never appears as the first token

and remains uncommon overall. It occurs somewhat more often later in GSM8K continuations after erroneous steps, while rates after correct steps remain near zero, showing that internally represented doubt rarely surfaces without intervention.

Table 8: Verbalization rates of doubt tokens without intervention (%).

Model	Dataset	First token		Next step		Anywhere	
		Err.	Corr.	Err.	Corr.	Err.	Corr.
Llama-3.1-8B-Instruct	Arithmetic	0.1	0.0	0.2	0.1	0.2	0.1
	Boolean	0.0	0.0	0.0	0.0	0.4	0.0
	GSM8K	1.4	0.0	4.3	0.0	10.1	1.1
Qwen2.5-7B-Instruct	Arithmetic	0.0	0.0	0.0	0.0	0.0	0.0
	Boolean	0.0	0.0	0.7	0.0	0.7	0.0
	GSM8K	0.0	0.0	5.9	0.7	7.4	0.7
Gemma-2-9B-IT	Arithmetic	0.0	0.0	0.0	0.0	0.0	0.0
	Boolean	0.0	0.0	0.0	0.0	0.0	0.0
	GSM8K	0.0	0.0	0.0	0.0	3.3	0.0

D Additional Results on Probe Intervention

Table 9 groups erroneous steps with incorrect greedy continuations by whether the probe intervention elicits a doubt-related first token. Across models and datasets, 25% of the 1,096 continuations that verbalize doubt recover, compared with 1% of the 1,731 that do not. Recovery is also absent in settings where the intervention never elicits doubt. Although this split is observational, the association is consistent with the probe intervention promoting correction by eliciting doubt.

Table 9: Recovery rate under the probe intervention after erroneous steps with incorrect greedy continuations, split by whether the intervened continuation begins with a doubt-related token (group size in parentheses; – denotes an empty group).

Model	Dataset	Doubt verbalized	Not verbalized
Llama-3.1-8B-Instruct	Arithmetic	0.23 (777)	0.01 (847)
	Boolean	0.64 (50)	0.00 (74)
	GSM8K	0.27 (115)	0.02 (66)
Qwen2.5-7B-Instruct	Arithmetic	–	0.03 (165)
	Boolean	0.33 (54)	0.00 (18)
	GSM8K	0.15 (68)	0.03 (34)
Gemma-2-9B-IT	Arithmetic	–	0.00 (245)
	Boolean	–	0.03 (175)
	GSM8K	0.16 (32)	0.00 (107)

E Additional Results on Non-Doubt Token Controls

Table 10 compares doubt and non-doubt J-lens interventions across all three models. For each model, we select the five non-doubt lexical tokens with the largest average gap in J-space selection rate between erroneous and correct steps, restricting candidates to words used in the model's own CoTs, and amplify them at the same layers and position as the doubt token. For each non-doubt token, we test three gains chosen with reference to the model-specific doubt-token gain, using $g \in \{3, 5, 8\}$ for Llama, $g \in \{8, 12, 16\}$ for Qwen, and $g \in \{3, 4, 5\}$ for Gemma. Each set includes the corresponding doubt-token gain. The first row for each model reproduces the doubt-token intervention from Table 3; the remaining rows report changes in final-answer accuracy relative to greedy continuation. The Verbalized columns give the fractions of continuations after erroneous steps that begin with the target token or any doubt-related token, averaged across datasets. Llama Arithmetic uses a random sample of 300 erroneous and 300 matched correct steps. Although the

non-doubt interventions often cause the target token to be verbalized, they rarely elicit doubt and yield small or inconsistent recovery, while sometimes degrading accuracy after correct steps. This contrast supports the specificity of doubt-related directions for self-correction.

Table 10: J-lens interventions on doubt and non-doubt tokens across models.

Model	Token	g	Arithmetic		Boolean		GSM8K		Verbalized	
			Err.	Corr.	Err.	Corr.	Err.	Corr.	Token	Doubt
Llama-3.1-8B-Instruct	␣BUT	5	+0.28	−0.10	+0.27	−0.02	+0.10	−0.13	0.96	0.96
		3	+0.03	−0.01	+0.07	−0.01	0.00	−0.03	0.29	0.00
	␣THEN	5	+0.07	−0.17	+0.05	−0.10	0.00	−0.08	0.88	0.00
		8	+0.06	−0.35	+0.06	−0.11	0.00	−0.14	0.96	0.00
		3	+0.01	0.00	0.00	0.00	−0.01	0.00	0.00	0.01
	␣multiplied	5	+0.01	−0.06	−0.04	−0.03	−0.04	−0.10	0.34	0.01
		8	+0.02	−0.28	−0.02	−0.04	−0.05	−0.21	0.76	0.01
		3	−0.05	−0.10	−0.04	−0.03	−0.04	−0.01	0.85	0.00
	␣Rounded	5	−0.07	−0.19	−0.04	−0.07	−0.07	−0.02	1.00	0.00
		8	−0.07	−0.23	−0.03	−0.04	−0.07	−0.01	1.00	0.00
		3	+0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.01
	␣divided	5	+0.01	−0.09	−0.02	−0.02	−0.03	−0.01	0.27	0.01
		8	−0.03	−0.19	+0.02	−0.02	−0.06	−0.08	0.48	0.00
		3	+0.01	0.00	0.00	0.00	+0.01	0.00	0.00	0.00
	␣times	5	+0.01	0.00	0.00	0.00	+0.01	0.00	0.00	0.00
		8	+0.01	0.00	+0.01	0.00	0.00	0.00	0.00	0.00
Qwen2.5-7B-Instruct	␣However	16	+0.17	−0.02	+0.15	0.00	+0.11	−0.02	1.00	1.00
		8	+0.01	+0.01	−0.03	−0.01	+0.02	−0.01	0.39	0.00
	␣Then	12	+0.01	+0.01	0.00	−0.01	+0.04	−0.01	0.95	0.00
		16	+0.01	+0.01	0.00	−0.01	+0.01	−0.01	0.99	0.00
		8	+0.04	0.00	+0.01	0.00	+0.04	−0.02	0.63	0.00
	␣Note	12	+0.04	−0.01	+0.05	0.00	+0.04	−0.01	1.00	0.00
		16	+0.06	−0.01	+0.05	0.00	+0.04	−0.01	1.00	0.00
		8	+0.01	0.00	−0.04	−0.01	−0.02	−0.01	0.66	0.00
	␣Final	12	+0.02	+0.01	+0.01	0.00	−0.03	−0.02	0.99	0.00
		16	+0.03	+0.01	+0.02	0.00	−0.03	−0.01	1.00	0.00
		8	+0.01	0.00	−0.03	−0.01	+0.01	−0.04	0.67	0.00
	␣gets	12	+0.02	−0.01	−0.02	0.00	−0.01	−0.07	0.88	0.00
		16	+0.02	0.00	0.00	−0.02	−0.01	−0.07	0.89	0.00
		8	−0.02	+0.01	−0.02	0.00	0.00	0.00	0.63	0.00
	␣So	12	−0.01	+0.01	0.00	0.00	−0.03	0.00	0.99	0.00
		16	−0.01	+0.01	−0.02	0.00	−0.03	0.00	1.00	0.00
Gemma-2-9B-IT	␣But	4	+0.04	−0.01	+0.06	−0.02	+0.04	−0.02	0.50	0.50
		3	+0.02	0.00	0.00	0.00	0.00	0.00	0.45	0.00
	␣finally	4	+0.01	−0.01	−0.02	0.00	−0.01	+0.01	0.77	0.00
		5	+0.01	0.00	0.00	0.00	−0.02	+0.01	0.86	0.00
		3	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00
	␣then	4	0.00	0.00	−0.03	0.00	0.00	0.00	0.51	0.00
		5	−0.01	0.00	−0.03	0.00	+0.01	0.00	0.73	0.00
		3	−0.04	0.00	+0.04	0.00	+0.02	0.00	0.47	0.00
	␣therefore	4	−0.07	0.00	−0.04	0.00	0.00	0.00	0.83	0.00
		5	−0.07	0.00	−0.05	0.00	+0.01	0.00	0.90	0.00
		3	−0.01	0.00	+0.02	0.00	0.00	0.00	0.04	0.00
	␣again	4	−0.01	0.00	+0.04	0.00	−0.01	0.00	0.49	0.00
		5	−0.01	0.00	+0.02	−0.03	0.00	0.00	0.62	0.00
		3	0.00	0.00	+0.02	−0.01	−0.01	0.00	0.14	0.00
	␣because	4	−0.02	0.00	+0.02	−0.01	+0.01	0.00	0.52	0.00
		5	−0.01	0.00	−0.02	−0.01	0.00	−0.01	0.80	0.00

F Additional Results on Text Insertion and Resampling

Table 11 compares the J-lens intervention with text insertion and resampling across all three models. Values are changes in final-answer accuracy relative to greedy continuation after erroneous and matched correct steps. The J-lens rows repeat our activation intervention, while Insert forces the indicated text at the start of the continuation before greedy decoding; the double-check condition inserts "Let me double-check the previous step." For Llama, we generate at least 16 additional tokens in this condition because the model otherwise terminates immediately. Across models, the J-lens intervention achieves recovery comparable to directly inserting doubt-related text, whereas resampling yields little change and the generic double-check instruction is less consistent.

Table 11: Comparison of J-lens intervention with text insertion and resampling.

Model	Method	Arithmetic		Boolean		GSM8K	
		Err.	Corr.	Err.	Corr.	Err.	Corr.
Llama-3.1-8B-Instruct	J-lens ␣BUT	+0.28	−0.10	+0.27	−0.02	+0.10	−0.13
	Insert ␣BUT	+0.25	−0.11	+0.25	0.00	+0.12	−0.16
	Insert ␣However	+0.33	−0.15	+0.27	−0.03	+0.09	−0.19
	Insert ␣Wait	+0.13	−0.08	+0.18	−0.03	+0.14	−0.23
	Insert ␣But	+0.23	−0.11	+0.22	−0.03	+0.12	−0.18
	Insert double-check	+0.11	−0.02	+0.16	−0.03	+0.09	−0.05
	Resample (T=0.7)	+0.01	−0.01	−0.01	−0.01	0.00	0.00
Qwen2.5-7B-Instruct	J-lens ␣However	+0.17	−0.02	+0.15	0.00	+0.11	−0.02
	Insert ␣BUT	+0.17	−0.03	+0.11	−0.01	+0.11	−0.05
	Insert ␣However	+0.16	−0.01	+0.10	0.00	+0.10	−0.04
	Insert ␣Wait	+0.18	−0.01	+0.14	0.00	+0.07	−0.02
	Insert ␣But	+0.20	−0.03	+0.12	−0.01	+0.05	−0.01
	Insert double-check	+0.08	+0.01	+0.14	0.00	+0.01	−0.01
	Resample (T=0.7)	+0.01	+0.01	+0.02	0.00	−0.01	0.00
Gemma-2-9B-IT	J-lens ␣But	+0.04	−0.01	+0.06	−0.02	+0.04	−0.02
	Insert ␣BUT	+0.05	−0.15	+0.09	−0.06	+0.04	−0.05
	Insert ␣However	+0.01	−0.22	+0.05	−0.12	+0.01	−0.04
	Insert ␣Wait	+0.02	−0.13	+0.08	−0.07	+0.03	−0.07
	Insert ␣But	+0.04	−0.04	+0.05	−0.03	+0.05	−0.06
	Insert double-check	+0.03	−0.01	+0.17	0.00	0.00	0.00
	Resample (T=0.7)	0.00	0.00	−0.01	0.00	−0.01	0.00

G Case Studies

Cases 1 and 2 show successful correction under J-lens and probe interventions, while the final case illustrates a failure caused by spurious doubt.

Case 1: Llama-3.1-8B-Instruct, Arithmetic

Problem. `( - ( - ( ( 7 + 1 ) * ( - 6 ) ) ) )` correct answer −48

CoT up to the erroneous step.

To evaluate this expression, we need to follow the order of operations (PEMDAS):
1. Evaluate the expression inside the innermost parentheses: (7 + 1) = 8
2. Multiply 8 by -6: 8 * -6 = -48
3. Multiply -48 by -1: -48 * -1 = 48

Internal state.

$c_{\ell,t}$ of ␣BUT at L18/L21/L26: 0.60/0.82/1.55 (correct-step median 0.47/0.67/1.24). J-space readout (L18): ␣BUT selected. Output head: ␣BUT rank 6, $P_{\text{doubt}} = 1.2 \times 10^{-3}$, $P_{\max} = 0.56$.

Greedy ($g = 1$) × 48

4. The final result is 48.

J-lens ($g^\star = 5$) ✓ −48
BUT, we have another - sign outside the parentheses, so we multiply 48 by -1: 48 * -1 = -48
The final answer is -48.
Probe ($g^\star = 5$) ✓ −48
but wait, there's another - sign outside the parentheses, so we multiply 48 by -1:
48 * -1 = -48
So the final answer is -48.
Insert ␣BUT ✓ −48
identical to the J-lens continuation.

Case 2: Gemma-2-9B-IT, GSM8K

Problem. James gets 10 new CDs. Each CD cost $15. He gets them for 40% off. He decides he doesn't like 5 of them and sells them for 40. How much money was he out? correct answer 50
CoT up to the erroneous step.
Here's how to solve the problem step-by-step:
1. Calculate the original cost:
* 10 CDs * $15/CD = $150
2. Calculate the discount amount:
* $150 * 40% = $60
3. Calculate the price James paid:
* $150 - $60 = $90
4. Calculate the value of the CDs he sold:
* 5 CDs * $15/CD = $75
Internal state.
$c_{\ell,t}$ of ␣But at L26/L29/L32: 26.7/36.9/37.4 (correct-step median 8.6/9.1/3.8). J-space readout (L26): ␣but (3rd of 25), ␣BUT, ␣unfortunately, ␣however selected. Output head: ␣BUT rank 19, $P_{\text{doubt}} = 6.9 \times 10^{-6}$, $P_{\max} = 0.99$.
Greedy ($g = 1$) × 15
5. Calculate James' net loss:
* $90 (cost) - $75 (selling price) = $15
Answer: James was out $15.
J-lens ($g^\star = 4$) ✓ 50
But remember, he sold them for $40, so he lost money on them.
5. Calculate James' total loss:
* $90 (initial cost) - $40 (selling price) = $50
Answer: James was out $50.
Probe ($g^\star = 4$) ✓ 50
However, he sold them for $40, so he lost money on them.
5. Calculate the total loss:
* $90 (initial cost) - $40 (selling price) = $50
Answer: James was out $50.
Insert ␣BUT ✓ 50
identical to the J-lens continuation, with **BUT** in place of **But**.

Failure case: Llama-3.1-8B-Instruct, GSM8K, J-lens intervention at $g^\star = 5$

Problem. A train has 172 people traveling on it. At the first stop 47 people get off and 13 more people get on, and at the next stop another 38 people get off. How many people are on the train? correct answer 100
CoT up to the intervened (correct) step.
Let's break it down step by step:
1. Initially, there are 172 people on the train.
2. At the first stop, 47 people get off, so the number of people remaining on the train is:
172 - 47 = 125
3. At the first stop, 13 people get on the train, so the new total number of people on the train is:
125 + 13 = 138
4. At the next stop, 38 people get off, so the final number of people on the train is:
138 - 38 = 100
Internal state.
$c_{\ell,t}$ of ␣BUT at L18/L21/L26: 0.23/0.30/0.90 (correct-step median 0.25/0.52/0.81).
Greedy ($g = 1$) ✓ 100
Therefore, there are 100 people on the train.
J-lens ($g^\star = 5$) × 113
BUT, we need to add the 13 people who got on at the first stop to the 100 people who are left after the second stop.
So, the correct answer is: 100 + 13 = 113