Oops, Wait: Token-Level Signals as a Lens into LLM Reasoning

Anonymous ACL submission

Abstract

The emergence of discourse-like tokens such as "wait" and "therefore" in large language models (LLMs) has offered a unique window into their reasoning processes. However, systematic analyses of how such signals vary across training strategies and model scales remain lacking. In this paper, we analyze token-level signals through token probabilities across various models. We find that specific tokens strongly correlate with reasoning correctness, varying with training strategies while remaining stable across model scales. A closer look at the "wait" token in relation to answer probability demonstrates that models fine-tuned on small-scale datasets acquire reasoning ability through such signals but exploit them only partially. This work provides a systematic lens to observe and understand the dynamics of LLM reasoning.

1 Introduction

Large language models (LLMs) have recently made remarkable progress across a wide range of reasoning-intensive tasks (Guo et al., 2025; Achiam et al., 2023; Touvron et al., 2023; Yang et al., 2024a,b, 2025a; Jiang et al., 2024; Team et al., 2025). Modern reasoning models explicitly generate a *reasoning trajectory*, which contains a step-by-step thinking process (Wei et al., 2022; Kojima et al., 2022; Achiam et al., 2023; Chen et al., 2025). These trajectories improve human interpretability (Lindsey et al., 2025), are leveraged during post-training to refine reasoning ability (Guo et al., 2025; Guha et al., 2025; Muennighoff et al., 2024; Ye et al., 2025), and, most importantly, lead to substantial gains on complex reasoning benchmarks.

Beyond overall reasoning performance, recent studies highlight the frequent emergence of discourse-like tokens within reasoning trajectories (Yang et al., 2025b; Liu et al., 2025; Qian et al., 2025). Tokens such as "wait", "therefore", and "alternatively" often appear at pivotal points in problem solving, functioning as anchors for reasoning structure and transitions toward the final answer. In human language, similar discourse markers play a central role in structuring arguments and using language fluently (Sun, 2013; Castro, 2009; Huneety et al., 2023; Stab and Gurevych, 2017), suggesting that token-level signals, how LLMs employ particular tokens during reasoning, may be closely tied to reasoning ability. Although a few works (Muennighoff et al., 2024; Zhang et al., 2025; Wang et al., 2025; Zhao et al., 2025b) attempt to leverage such patterns to improve reasoning, systematic comparisons remain limited. Understanding token-level signals offers a concrete lens for analyzing how reasoning behavior varies with different training strategies and model scales.

In parallel, a milestone study, s1 (Muennighoff et al., 2024), showed that supervised fine-tuning (SFT) with only 1K curated reasoning examples can enhance the reasoning ability of LLMs. This dataset sampled DeepSeek-R1 trajectories, effectively positioning DeepSeek-R1 as the teacher from which other models learn. In human language learning, prior work suggests that acquiring discourse markers is a central step in developing persuasive and logical argumentation (Sun, 2013). By analogy, models fine-tuned on teacher trajectories must also acquire and internalize token-level signals, especially discourse-like tokens such as "wait", "therefore", or "alternatively", to transfer reasoning ability effectively. This raises a sharper question: *do small-scale SFT datasets successfully inherit token-level signals in the way humans acquire discourse markers or impose limitations?*

In this paper, we investigate token-level signals in LLM reasoning through two in-depth analyses: the first examines how next-token generation probabilities of tokens differ depending on whether the final answer is correct or incorrect, and the second explores how the occurrence of the "wait" token affects the probability of producing correct answers.

We show that (1) specific tokens are strongly associated with correctness while others correlate with incorrect reasoning, (2) these associations vary systematically across training strategies and model scales, and (3) small-scale supervised fine-tuning transfers token signals but only partially, leading to differences in reasoning performance.

Our analyses also provide practical insights for developing token-level steering, ensemble methods, and post-training strategies. In addition, these token-level signals can be applied in ensemble settings by identifying and filtering unreliable generations. Moreover, although small-scale SFT has limited capacity to instill robust token-level signals, our findings suggest that emphasizing important discourse tokens can still improve reasoning performance, even under constrained data or training budgets. Overall, our analyses offer concrete directions toward more controllable and effective reasoning.

2 Related Work

Anthropomorphic expressions and discourse markers in reasoning. Recent studies have investigated how such markers appear in LLM reasoning. Yang et al. (2025b) analyzed external signals interpreted as anthropomorphic expressions associated with “aha-moments”, while Liu et al. (2025) discussed the emergence of reflective patterns such as “wait” or “hmm” as discourse markers in R1-style reasoning models. Qian et al. (2025) further identified that so-called thinking tokens correspond to peaks of mutual information within reasoning trajectories. These works suggest the link between linguistic discourse markers and reasoning behavior, but they primarily focus on qualitative or within-model observations. In contrast, we present a systematic token-level analysis across models trained with different recipes and scales, revealing how discourse-like tokens relate to correctness and vary across training conditions.

“Wait” in reasoning of LLMs. A line of research has explored manipulating reasoning dynamics through explicit control tokens such as “wait”. Muennighoff et al. (2024) introduced test-time scaling via controlled reasoning extension with “wait”. Building on this idea, Zhao et al. (2025b) linked “wait” to activation patterns and proposed an activation control method, while Zhang et al. (2025) modeled reasoning as a balance between slow and fast thinking mediated by the “wait” token. In parallel, Wang et al. (2025) reported that removing “wait”-like tokens can even improve efficiency in some settings. While these studies highlight the importance of “wait” in shaping reasoning behavior, they primarily use it as control rather than as an analytical lens. Our work shifts the focus to analysis of how “wait” functions within reasoning trajectories, quantifying model-level differences and its partial transfer through small-scale SFT.

3 Token-level Signals in Reasoning

In this section, we present a systematic analysis of token-level signals in reasoning trajectories. Beyond simple frequency counts, we quantify these signals by computing token probabilities, defined as the average probability of tokens generated immediately after “\n\n”. This allows us to compare how tokens behave across correct versus incorrect reasoning, across various models, and we also examine their relation to model confidence. By examining these dimensions, we identify which tokens consistently act as markers of successful reasoning and which are associated with errors, providing a quantitative basis for understanding how training strategies and model scales shape reasoning ability.

3.1 Computing token probabilities

Figure 1 illustrates how the token probability is computed by collecting the softmax probabilities of the next token after “\n\n” and averaging them. Let $X = \{x_1, x_2, \ldots, x_T\}$ denote the token set in a trajectory, and let $I_X = \{i : x_i = \text{“\textbackslash n\textbackslash n”},\ x_i \in X\}$ be a set of positions where “\n\n” occurs. The average token probability for trajectory X is $p_X = \frac{1}{|I_X|}\sum_{i \in I_X} p(x_{i+1})$, where $p(x_{i+1})$ denotes the probability assigned by the model to token x_{i+1}.

We further compute aggregated statistics across multiple trajectories. Among all trajectories in a dataset $D = \{X_1, X_2, ...X_N\}$, we use two subsets: correct-answer $D_{true} = \{X_i : \text{Answer}(X_i) = true\}$ and incorrect-answer $D_{false} = \{X_i : \text{Answer}(X_i) = false\}$. Specifically, we define the mean token probability over correct- and incorrect-answer trajectories as

$$\bar{p}_* = \frac{1}{|D_*|} \sum_{X \in D_*} p_X, \quad (1)$$

where $* \in \{true, false\}$ for correct p_{true} or incorrect p_{false} mean probabilities, respectively.

We conduct experiments on AIME24 (OpenAI, 2024), GPQA-D (Rein et al., 2024), and MATH-500 (Lightman et al., 2023). We use the following models: DeepSeek-R1-distill-Qwen-32B (Guo

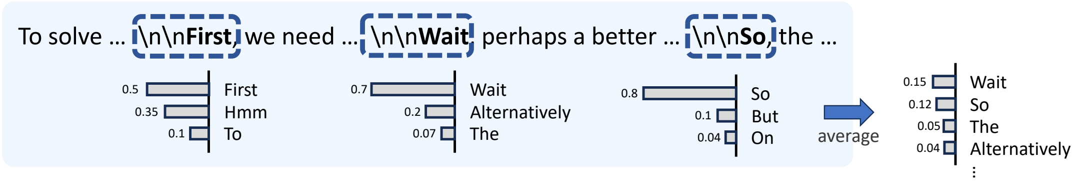


Figure 1: **Overview of how token probabilities are collected.** We extract next-token probability distributions specifically after "\n\n", which serve as natural segmentation points in the trajectories. This enables us to capture latent token-level signals beyond simple frequency counts, reflecting how strongly the model intends to generate particular tokens even when they are not actually selected during generation. Such token-probability measures enable a fine-grained comparison of token-level signals across models.

Token	$\bar{p}_{true}(t)$	$\bar{p}_{false}(t)$	$\Delta(t)$
I	4.0%	1.3%	+2.7%
Therefore	4.1%	1.6%	+2.5%
The	5.0%	3.1%	+1.9%
Let	3.1%	1.4%	+1.7%
Now	3.4%	1.8%	+1.6%
So	11.9%	10.4%	+1.5%
But	3.7%	7.2%	-3.5%
Alternatively	3.7%	9.0%	-5.2%
Wait	15.4%	25.8%	-10.4%

Table 1: **Tokens with statistically significant differences** (t-test, $p < 0.05$) between correct and incorrect trajectories for **DeepSeek-R1-distill-Qwen-32B.** Tokens are sorted by $\Delta(t)$.

Token	$\bar{p}_{true}(t)$	$\bar{p}_{false}(t)$	$\Delta(t)$
Therefore	7.6%	3.6%	+4.0%
So	5.6%	3.7%	+1.8%
Now	2.0%	1.2%	+0.8%
Let	2.4%	1.8%	+0.6%
The	5.5%	8.5%	-3.1%
Alternatively	7.8%	17.8%	-9.9%

Table 2: **Tokens with statistically significant differences** (t-test, $p < 0.05$) between correct and incorrect trajectories for **QwQ-32B**. Tokens are sorted by $\Delta(t)$.

et al., 2025), QwQ-32B (Yang et al., 2024b, 2025a), s1.1-32B (Muennighoff et al., 2024), and s1-32B (Muennighoff et al., 2024). Note that DeepSeek-R1-distill-Qwen-32B, s1.1-32B, and s1-32B are post-trained on the same base model, Qwen2.5-32B-Instruct (Yang et al., 2024b) with different settings, and that QwQ-32B is also based on Qwen2.5-32B. To analyze model-size effects, we additionally evaluate DeepSeek-R1-distill-Qwen-14B and DeepSeek-R1-distill-Qwen-7B. For brevity, we refer to DeepSeek-R1-distill-Qwen-32B as R1-32B in the remainder of this paper. Further details are provided in appendix A and B.

For each model, we perform a t-test on token probability distributions to assess whether they differ significantly between correct and incorrect reasoning. In addition, we prompt the models to generate confidence estimates, allowing us to examine the interaction between correctness, confidence, and token-level probabilities.

3.2 Token-level signals with answer correctness

Tables 1 and 2 present representative examples of how token probabilities vary with reasoning correctness in R1-32B and QwQ-32B. For both models, we report tokens whose probabilities differ significantly between correct and incorrect answers according to a t-test ($p < 0.05$). As shown, some tokens exhibit large differences (e.g., "wait" in R1-32B, "alternatively" in QwQ-32B), while others show relatively smaller gaps (e.g., "so" in R1-32B, "let" in QwQ-32B). Moreover, the same token (e.g., "the") can behave differently across models, suggesting deeper variability in token-level signals.

Training recipes change token-level signals. To further examine how training recipes influence token-level patterns, we compare correct- and incorrect-associated tokens across models that share the same architectural baseline (Qwen2.5). Table 3 presents the lists of correct- and incorrect-associated tokens across different models. Note that tokens in Table 3 are arranged in descending order of $|\Delta(t)|$, which indicates the strength of the correlation. Additionally, tokens with negative or contrastive roles, such as "however", "but", "consider", and "alternatively", tend to appear as incorrect-associated tokens. In contrast, tokens that convey positive progression, such as "therefore", "so", and "let", are typically included among correct-associated tokens. Interestingly, the associated tokens, i.e., token-level signals, vary substantially across models. R1-32B, s1.1-32B, and s1-32B all share the same base architecture, Qwen2.5-32B-Instruct, yet their signals diverge depending on the training data and supervision used. While s1.1-32B, trained on DeepSeek-R1 trajec-

Model	Type	Associated Tokens
R1-32B	Correct	I, Therefore, The, Let, Now, So
	Incorrect	Wait, Alternatively, But
QwQ-32B	Correct	Therefore, So, Now, Let
	Incorrect	Alternatively, The
s1.1-32B	Correct	Therefore, So, First
	Incorrect	Alternatively, Wait
s1-32B	Correct	The, Conf, Final
	Incorrect	*, Consider

Table 3: **Correct- and incorrect-associated tokens across various models.** Tokens within each group are sorted in descending order of $|\Delta(t)|$.

Model	Type	Associated Tokens
32B	Correct	I, Therefore, The, Let, Now, So
	Incorrect	Wait, Alternatively, But
14B	Correct	Therefore, Let, Now, So, I, The
	Incorrect	Wait, Alternatively, But
7B	Correct	The, Now, Therefore, Let, I
	Incorrect	Wait, Alternatively, Hmm, But

Table 4: **Correct- and incorrect-associated tokens across different model scales** (7B, 14B, 32B) of the DeepSeek-R1-distill-Qwen series. Tokens within each group are sorted in descending order of $|\Delta(t)|$.

Model	Pearson	Spearman
R1-32B	0.9899^{**}	1.0000^{***}
QwQ-32B	0.8647^{*}	0.8286^{*}
s1.1-32B	0.9076^{*}	1.0000^{***}
s1-32B	0.7971	0.9000^{*}

Table 5: **Correlation between the correct-incorrect token probability gap and model confidence**. Both Pearson and Spearman correlations are reported (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$).

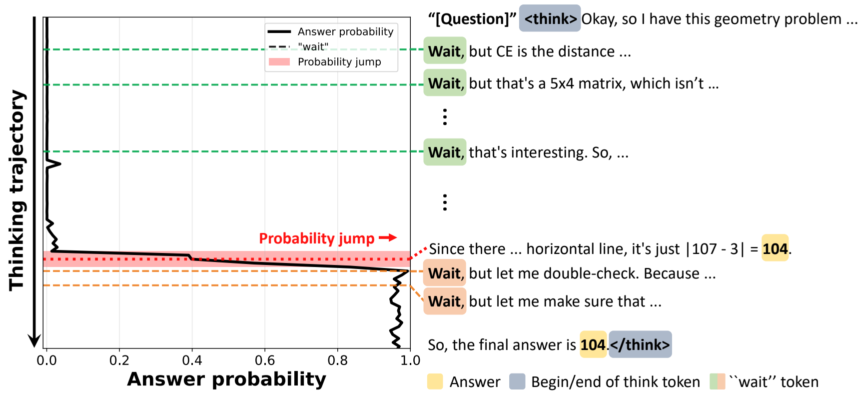
tories, shows strong similarity to R1-32B, s1-32B, trained on Gemini-derived reasoning traces, exhibits markedly different associations. Similarly, QwQ-32B, though based on the same backbone (Qwen2.5-32B), shows further deviations due to its extensive combination of reinforcement learning and SFT. These observations suggest that token-level signals are determined primarily by training strategies rather than model architecture.

Model scales preserve token-level signals. We compare three models of different sizes from the same series, DeepSeek-R1-distill-Qwen-32B, 14B, and 7B, as shown in Table 4. Interestingly, the sets of correct- and incorrect- associated tokens remain consistent across scales, with only minor variations in their relative importance ($|\Delta(t)|$). This consistency suggests that token-level patterns are largely independent of model capacity and are instead governed by the shared training recipe.

Token-level signals and confidence. Beyond correctness, we further examine how token-level probability differences relate to model confidence. For each trajectory, we define the correct–incorrect token probability gap as the difference between the summed probabilities of correct-associated and incorrect-associated tokens, and examine its correlation with the model's self-reported confidence. As shown in Table 5, both Pearson and Spearman coefficients show strong positive correlations across all models, indicating that larger gaps are closely aligned with higher confidence. These findings indicate that token-level probability patterns not only reflect correctness but are also significantly associated with the self-reported confidence of the model, supporting the reliability of such token-level analyses as meaningful behavioral indicators.

Summary of our findings. We investigate token-level signals in models, focusing on which tokens serve as indicators of reasoning success and failure. Our analysis shows that progression tokens (e.g., "therefore", "so") are consistently associated with correctness, while contrastive tokens (e.g., "wait", "alternatively") are associated with incorrectness (Table 1-2). Importantly, these token-level signals vary across training recipes (Table 3) but remain stable across model sizes (Table 4), suggesting that they are influenced more by training recipes than by capacity. Moreover, token-level probability patterns correlate strongly with model confidence, supporting their reliability as behavioral indicators (Table 5). Interestingly, in both R1-32B and s1.1-32B, which are trained on DeepSeek-R1 trajectories with the same backbone, "wait" consistently emerges as an incorrect-associated token. The effect is strongest in R1-32B, with the highest $\Delta(t)$, but is noticeably weaker in s1.1-32B. This observation motivates a closer examination of why such closely related models differ in their use of "wait".

Takeaway from §3. *Token-level signals reveal distinct reasoning dynamics:* Several tokens are strongly tied to reasoning correctness, showing how models organize reasoning cues and how supervision, rather than capacity, shapes reasoning behavior.



"[Question]" <think> Okay, so I have this geometry problem ...

Wait, but CE is the distance ...

Wait, but that's a 5x4 matrix, which isn't ...

⋮

Wait, that's interesting. So, ...

⋮

Since there ... horizontal line, it's just |107 - 3| = 104.

Wait, but let me double-check. Because ...

Wait, but let me make sure that ...

So, the final answer is 104.</think>

Figure 2: Changes in **answer probabilities** and **emergence of "wait" and its subsequent tokens** along the thinking trajectory. Horizontal dashed lines indicate where "wait" is generated, and the probability jump region is highlighted, with the red dashed line marking the point of maximum increase. Expressions following "wait" differ depending on whether they occur before or after the probability jump, with earlier instances more often extending the reasoning and later instances serving as re-checks. Note that answer probabilities are computed at the token level, although the figure is visualized in a line-based format.

4 "Wait" in Reasoning Trajectories

In this section, we take a closer look at the role of "wait" and examine how it relates to differences in reasoning performance. We first analyze how "wait" relates to a model's progression toward the final answer by truncating reasoning trajectories and prompting answers from incomplete reasoning. This allows us to trace how the probability of producing a correct answer evolves, revealing that it does not increase smoothly but instead exhibits sharp jumps at key points. These jumps are often preceded or followed by "wait", serving as a trigger or a self-checking step (Figure 2). Based on these, we further compare the token-level signals associated with "wait" between R1-32B and s1.1-32B to examine the effect of small-scale SFT.

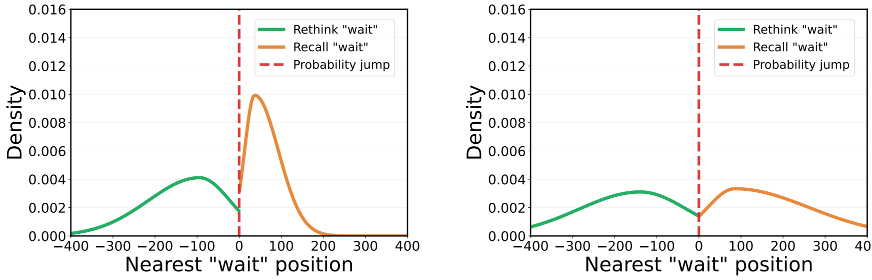
4.1 Answer probabilities

To track confidence in a particular answer during reasoning, we define and compute the answer probability as the probability that the model would generate the correct final answer. We obtain this probability at intermediate points of the thinking trajectory by truncating it at fixed intervals of 10 tokens and prompting the model to directly generate the final answer. To ensure consistency, we insert the model-specific answer delimiter token, "<|im_start|>answer" for s1 and "</think>" for DeepSeek, followed by the prefix "Final answer: \boxed{". This setup ensures that the model produces the final answer explicitly.

We conduct this analysis using s1.1-32B and R1-32B on 30 problems from AIME24. For each problem, we generate three responses: one with a temperature of 0 and two with a temperature of 0.6. Since all AIME answers are numerical, the probability of a correct answer is computed from the predicted distribution over digits. If the correct answer consists of multiple digits, we condition the generation by fixing the preceding digits and compute the probability of each subsequent digit sequentially. Formally, for an answer represented as a sequence of digits $a = (d_1, d_2, \ldots, d_T)$, the probability is given by

$$P(a \mid \text{prompt}) = \prod_{t=1}^{T} P(d_t \mid d_{<t}, \text{prompt}). \quad (2)$$

Intuitively, one might expect the answer probability to rise gradually as the reasoning trajectory progresses. We instead observe that the probability often exhibits sudden leaps rather than following a smooth trend. To capture this phenomenon, we define a *probability jump* as the point in the trajectory where the increase in answer probability is maximized. We detect such a jump by sliding a window along each probability trajectory and computing, for each token position t, the difference between the average probability over the four preceding steps and the four following steps. The position t that maximizes this difference is designated as the *probability jump*. Figure 2 illustrates an example trajectory with a highlighted jump area. Notably, such sharp probability jumps are consistently observed in all cases where the model arrives at the correct answer.



(a) DeepSeek-R1-distill-Qwen-32B

(b) s1.1-32B

Figure 3: Distribution of the **relative positions of the nearest rethink and *recall "wait"* token to the probability jump**. For each reasoning trajectory, exactly one rethink and one recall token are selected, which might cause or follow the probability jump. An asymmetric Gaussian curve is fitted to each distribution.

4.2 Role of "wait"

Figure 2 illustrates that "wait" tokens appear before and after a probability jump. To systematically analyze this behavior, we classify every occurrence of "wait" in the reasoning trajectory relative to the jump point. We introduce two categories: *rethink "wait"* and *recall "wait"*. A *rethink "wait"* is any "wait" token that appears before the probability jump, typically used to extend or reconsider the ongoing reasoning. Examples include phrases such as *"Wait, but in this case ..."* or *"Wait, but actually ..."*, where the token pushes the reasoning forward by exploring alternatives, and *"Wait, that's interesting. So ..."*, where the token extends the reasoning by building on the current line of thought. In contrast, a *recall "wait"* is any occurrence of "wait" after the jump, generally produced when the model has already reached the solution and is double-checking or summarizing its result. For instance, it appears in forms like *"Wait, but let me double-check ..."* or *"Wait, but let me think again ..."*, signaling verification or restatement. In other words, we label each "wait" token based on whether it comes prior to the confidence leap (rethink) or following it (recall).

We then analyze the patterns of *rethink* and *recall "wait"*s across two models: s1.1-32B, trained on a small generated dataset (s1K-1.1), and R1-32B. Our analysis proceeds along four dimensions: their positions relative to probability jumps, the magnitude of probability increases following *rethink "wait"*, their overall frequency, and their occurrence in incorrect samples. Together, these four perspectives provide a comprehensive view of how "wait" tokens operate in reasoning trajectories and how their usage differs across models. Note that these analyses, except for incorrect sample analysis, are conducted on thinking trajectories that reach correct answers, since probability jumps toward correct answers are not observed in incorrect cases.

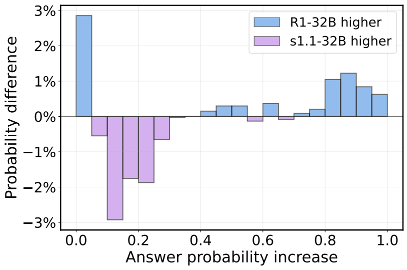
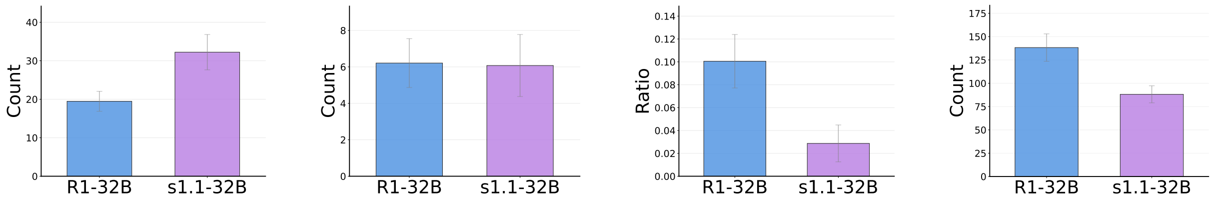


Figure 4: **Difference in answer probability increase distributions** following rethink "wait" tokens between DeepSeek-R1-distill-Qwen-32B and s1.1-32B.

Nearest "wait" to the probability jump. We hypothesize that the "wait" closest to the probability jump is the most relevant "wait". In the case of *rethink "wait"*, it might trigger a reasoning step that makes the probability jump. Otherwise, the probability jump might cause the closest *recall "wait"*, encouraging the model to review its reasoning and confirm the answer. We exclude the "wait" beyond 400 tokens from the probability jump, since they are too far from the jump point to consider them relevant. Based on this hypothesis, we compare the closest rethink and *recall "wait"* for R1-32B and s1.1-32B. The distributions of the closest tokens are illustrated in Figure 3. While the distributions of *rethink "wait"* are similar across the two models, the *recall "wait"* shows substantially different distributions between R1-32B and s1.1-32B. This implies that although the s1.1-32B learns the position of *rethink "wait"* from the small dataset enough to mimic R1-32B, its revisiting process through *recall "wait"* differs, which might be related to their reasoning pattern and performance gap.

Answer probability increase after "wait". We measure the amount of answer probability increase after every *rethink "wait"* tokens to evaluate the success ratio of rethinking. We report the maxi-



(a) Total *rethink "wait"* (b) Total *recall "wait"* (c) Success ratio (d) Incorrect samples

Figure 5: **Statistics of *rethink* and *recall "wait"* tokens** across DeepSeek-R1-distill-Qwen-32B and s1.1-32B. (a)–(d) show the number of *rethink "wait"* and *recall "wait"* tokens, the ratio of *rethink "wait"* tokens followed by a significant probability increase, and the total number of "wait" tokens in incorrect trajectories, respectively.

mum probability increases within a 384-token window after the "wait" token. Figure 4 shows the difference in answer probability increase distributions between R1-32B and s1.1-32B models. In case of s1.1-32B, the answer probability increases are concentrated in the low increase region compared to R1-32B. A few *rethink "wait"* make the probability jump close to 100%, but only in a small portion 1.6%. Otherwise, R1-32B has a separate increase pattern. While it has a lower percentage (around 0%) of small probability increases than s1.1-32B, it demonstrates a strong high probability increase (> 80%). Figure 5c reports the ratio of probability increase exceeds 80% after applying question-level normalization. R1-32B *rethink "wait"* is about four times more likely to make a probability jump (increase > 0.8) compared to s1.1-32B. Overall, these analyses indicate that "wait" tokens in R1-32B have superior effects compared with those in s1.1-32B.

Quantitative statistics for "wait". We compare "wait" in R1-32B and s1.1-32B with quantitative numbers aggregated across all questions. Figure 5 reports the numbers based analyses. As shown in Figure 5a, s1.1-32B uses more "wait" tokens than R1-32B in rethink cases, about 1.5 times more. As indicated in Figure 4, this suggests an overuse of *rethink "wait"* in s1.1-32B, which occurs more often but is less tied to probability increases. In contrast, the number of *recall "wait"* is similar between the two models, but, as shown in Figure 3, it exhibits a weaker association with probability jumps. Figure 5d visualizes the number of "wait" in incorrect samples that are excluded from rethink and recall analyses. Interestingly, R1-32B employs more "wait" in the incorrect samples than s1.1, in contrast to its usage in correct samples. R1-32B appears to use more "wait" than s1.1-32B when it cannot find a path to the answer, but it does not lead to overuse due to its superior "wait" success ratio in Figure 5c.

Summary of our findings. We investigate the difference in the use of "wait" between s1.1-32B and R1-32B. R1-32B demonstrates superior usage of "wait" compared to s1.1-32B in the following aspects: precise position of *recall "wait"* (Figure 3), answer probability increases (Figure 4), and the numbers of "wait" tokens (Figure 5a and Figure 5b). Overall, we conclude that s1.1-32B learns to use "wait" tokens from the small generated dataset, s1K-1.1, which shows some effectiveness but is insufficient to capture the detailed usage and effects of "wait". s1.1-32B tends to overuse "wait" compared to R1-32B. In terms of probability jumps, R1-32B's "wait" outperforms that of s1.1-32B with a substantially higher success rate. We conjecture that these limitations are a drawback of training with the small generated dataset.

Takeaway from §4. *Small-scale SFT leaves token-level signals underdeveloped*: Analyzing "wait" around answer probability jumps shows that s1.1-32B fails to exploit these tokens as effectively as R1-32B, underscoring the limits of small-dataset training in transferring reasoning signals.

5 Discussion

In this section, we discuss how our analyses inform token-level steering and post-training strategies for reasoning. These discussions extend our analyses and suggest practical directions for achieving effective reasoning.

Insights on token suppression. In Section 3, we find that several tokens are strongly associated with model correctness and confidence. This intuitively raises the question of whether manipulating these tokens, particularly incorrect-associated ones, can steer model performance. We examine this question by evaluating three models, s1.1-32B, R1-32B, and QwQ-32B, across three settings: suppressing correct-associated tokens, incorrect-associated to-

	Baseline	Correct	Incorrect	All
R1-32B	81.0%	80.2%	77.3%	78.5%
s1.1-32B	70.2%	74.9%	71.2%	70.9%
QwQ-32B	74.7%	73.2%	70.1%	68.0%

Table 6: **Effect of associated-token suppression on model performance.** Results are averaged over AIME24, GPQA-Diamond, and MATH benchmarks. "Baseline", "Correct", "Incorrect", and "All" refer to no suppression, suppression of correct-associated tokens, suppression of incorrect-associated tokens, and suppression of all associated tokens, respectively.

	Pass@1	Maj. V.	DeepConf	T-Gap (ours)
R1-32B	50.0%	**60.0%**	**60.0%**	**60.0%**
s1.1-32B	43.3%	53.3%	**56.7%**	**56.7%**
QwQ-32B	56.7%	63.3%	66.7%	**70.0%**

Table 7: **Ensemble using token-level signals.** Results are on the AIME24 benchmark. "Maj. V." and "T-Gap" refer to the majority voting strategy and the ensemble strategy based on the correct–incorrect token probability gap proposed in this work, respectively.

kens, and both groups simultaneously.

As shown in Table 6, the experimental results contradict the simple assumption. While suppressing correct-associated tokens sometimes leads to slight improvements (e.g., in s1.1-32B) or yields comparable performance, suppressing incorrect-associated tokens or both groups consistently degrades results. This pattern suggests that incorrect-associated tokens play a crucial role in maintaining reasoning ability. The observation is consistent with Section 3, where tokens exhibiting the largest $|\Delta(t)|$ were predominantly incorrect-associated. This outcome suggests that steering methods incorporating token suppression could be more effective by exploiting model-specific token-level signals identified in our analyses for reasoning control.

Insights on ensemble. Beyond steering individual generations, token-level signals can also give information to select reliable samples across multiple trials. Majority voting, which is a representative ensemble method, is one of the most common strategies for improving the performance of LLMs and enhancing reliability. Previous work (Fu et al., 2025) has shown that group-level confidence can be used to weight or filter answers from multiple trials, achieving performance gains beyond simple majority voting. Here, we instead utilize the correct–incorrect token probability gap, which exhibits a strong correlation with model confidence. We treat responses with low gap values as unreliable and exclude the bottom 20% of samples based on this metric during ensembling. All experiments are conducted on AIME24, where ensembling is performed over 32 sampled responses.

Table 7 presents the ensemble results. All ensemble methods outperform pass@1. Notably, our approach using correct-incorrect token probability gap achieves the best performance across all three models. This observation suggests that token-level signals can provide insights for enhancing model performance beyond indicating simple correlation with confidence or internal model signals.

Insights on post-training for reasoning. In Section 4, we demonstrate both the potential and limitations of small-scale SFT. Our findings suggest that models often fail to fully exploit discourse-level tokens that structure reasoning. As discussed in Section 6, human language learning also relies on acquiring discourse markers to organize and refine logical arguments; analogously, LLMs may require targeted learning on these markers to develop robust reasoning behaviors. Similarly, recent work on safety alignment (Zhao et al., 2025a) demonstrates that emphasizing refusal-related tokens enhances model robustness during post-training. We conjecture that token-centric supervision, focusing on tokens highly associated with reasoning correctness, could serve as an effective strategy for reasoning-oriented post-training.

6 Conclusion

In this work, we have analyzed token-level signals to understand how large language models acquire and apply reasoning. Progressive tokens such as "therefore" and "so" are strongly linked to correct reasoning, whereas contrastive ones like "wait" and "alternatively" often accompany incorrect reasoning. These patterns remain stable across model sizes but vary with training recipes, suggesting that supervision plays a greater role than scale in shaping reasoning behavior. Further analysis of the "wait" token shows its dual role as both a trigger for probability shifts and a cue for self-checking. It also reveals that small-scale SFT captures token-level signals only partially. Our analyses of token-level signals provide valuable insights into token-level steering, ensemble methods, and post-training strategies for reasoning, suggesting future directions for controllable and effective reasoning.

Limitations

While this study focuses on training strategies and model scales, our analyses primarily rely on open-source models from the Qwen series. Future work could extend this framework to other base architectures such as LLaMA (Touvron et al., 2023) or Mixtral (Jiang et al., 2024) to test its generality. Because our method requires access to softmax outputs and full reasoning trajectories, it is less applicable to closed-source reasoning models such as GPT (Achiam et al., 2023) or Gemini (Comanici et al., 2025). We also analyze three representative reasoning benchmarks that emphasize natural-language reasoning, where discourse markers naturally play a central role. Code-generation settings (Quan et al., 2025; Penedo et al., 2025; Jain et al., 2024) may exhibit different types of reasoning signals beyond discourse markers, which remain an interesting direction for future exploration.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Claudia Marcela Chapetón Castro. 2009. The use and functions of discourse markers in efl classroom interaction. *Profile: Issues in Teachers' Professional Development*, (11):57–77.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *arXiv preprint arXiv:2508.15260*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, and 1 others. 2025. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*.

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

Anas Huneety, Asim Alkhawaldeh, Bassil Mashaqba, Zainab Zaidan, and Abdallah Alshdaifat. 2023. The use of discourse markers in argumentative compositions by jordanian efl learners. *Humanities and Social Sciences Communications*, 10(1):1–8.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

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

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, and 8 others. 2025. On the biology of a large language model. *Transformer Circuits Thread*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. 2024. s1: Simple test-time scaling. In *Workshop on Reasoning and Planning for Large Language Models*.

OpenAI. 2024. Learning to reason with llms. OpenAI blog.

Guilherme Penedo, Anton Lozhkov, Hynek Kydlíček, Loubna Ben Allal, Edward Beeching, Agustín Piqueres Lajarín, Quentin Gallouédec, Nathan Habib, Lewis Tunstall, and Leandro von Werra. 2025. Codeforces. `https://huggingface.co/datasets/open-r1/codeforces`.

Chen Qian, Dongrui Liu, Haochen Wen, Zhen Bai, Yong Liu, and Jing Shao. 2025. Demystifying reasoning dynamics with mutual information: Thinking tokens are information peaks in llm reasoning. *arXiv preprint arXiv:2506.02867*.

Shanghaoran Quan, Jiaxi Yang, Bowen Yu, Bo Zheng, Dayiheng Liu, An Yang, Xuancheng Ren, Bofei Gao, Yibo Miao, Yunlong Feng, and 1 others. 2025. Codeelo: Benchmarking competition-level code generation of llms with human-comparable elo ratings. *arXiv preprint arXiv:2501.01257*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3).

Wei Sun. 2013. The importance of discourse markers in english learning and teaching. *Theory & Practice in Language Studies (TPLS)*, 3(11).

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Chenlong Wang, Yuanning Feng, Dongping Chen, Zhaoyang Chu, Ranjay Krishna, and Tianyi Zhou. 2025. Wait, we don't need to" wait"! removing thinking tokens improves reasoning efficiency. *arXiv preprint arXiv:2506.08343*.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Shu Yang, Junchao Wu, Xin Chen, Yunze Xiao, Xinyi Yang, Derek F Wong, and Di Wang. 2025b. Understanding aha moments: from external observations to internal mechanisms. *arXiv preprint arXiv:2504.02956*.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

Dongkeun Yoon, Seungone Kim, Sohee Yang, Sunkyoung Kim, Soyeon Kim, Yongil Kim, Eunbi Choi, Yireun Kim, and Minjoon Seo. 2025. Reasoning models better express their confidence. *arXiv preprint arXiv:2505.14489*.

Junyu Zhang, Runpei Dong, Han Wang, Xuying Ning, Haoran Geng, Peihao Li, Xialin He, Yutong Bai, Jitendra Malik, Saurabh Gupta, and 1 others. 2025. Alphaone: Reasoning models thinking slow and fast at test time. *arXiv preprint arXiv:2505.24863*.

Xuandong Zhao, Will Cai, Tianneng Shi, David Huang, Licong Lin, Song Mei, and Dawn Song. 2025a. Improving llm safety alignment with dual-objective optimization. In *Forty-second International Conference on Machine Learning*.

Zekai Zhao, Qi Liu, Kun Zhou, Zihan Liu, Yifei Shao, Zhiting Hu, and Biwei Huang. 2025b. Activation control for efficiently eliciting long chain-of-thought ability of language models. *arXiv preprint arXiv:2505.17697*.

Appendix

A Experimental Settings

Models. All models used in this study are publicly available open-source checkpoints released on HuggingFace under permissive licenses. We use six reasoning-oriented LLMs: DeepSeek-R1-distill-Qwen-32B[1], DeepSeek-R1-distill-Qwen-14B[2], DeepSeek-R1-distill-Qwen-7B[3], s1.1-32B[4], s1-32B[5], and QwQ-32B[6]. Licenses are MIT (DeepSeek-R1), Apache 2.0 (s1.1-32B and s1-32B), and Qwen Community License (QwQ-32B). Each model is used solely for research purposes.

Datasets and evaluation. We evaluate models on three reasoning-focused benchmarks: AIME24[7], GPQA-Diamond[8], and MATH[9]. Each dataset is used solely for evaluation purposes for scientific research. All evaluations are performed using the `lm-evaluation-harness` codebase (Gao et al., 2024)[10] and the `s1` codebase[11], both of which were adapted for our analysis.

All of these models and datasets are utilized for studying the reasoning of LLMs. The models (e.g., Qwen2.5, DeepSeek-R1, and QwQ series) were pretrained on multilingual corpora that include both English and Chinese data. However, all analyses and evaluations in this study were conducted exclusively on English-language benchmarks (AIME24, GPQA-Diamond, and MATH-500).

Chat template for self-reported confidence. To obtain self-reported confidence in Section 3, we use the chat template shown below, following the design proposed by Yoon et al. (2025).

Chat Template

First, solve the following math problem efficiently and clearly.
Then, thoroughly assess your confidence in that answer by evaluating your thinking process so far.
Finally, classify your confidence into one of the following classes based on how likely your answer is to be correct:
- "Almost no chance" (0.0–0.1)
- "Highly unlikely" (0.1–0.2)
- "Chances are slight" (0.2–0.3)
- "Unlikely" (0.3–0.4)
- "Less than even" (0.4–0.5)
- "Better than even" (0.5–0.6)
- "Likely" (0.6–0.7)
- "Very good chance" (0.7–0.8)
- "Highly likely" (0.8–0.9)
- "Almost certain" (0.9–1.0)

Each category reflects the probability that your answer is correct.
The last line of your response should be of the following format: Therefore, the final answer is: $\boxed {{ANSWER}}$, Confidence: $CLASS. I hope it is correct. (without quotes) where ANSWER is just the final number or expression that solves the problem and CLASS is one of the names (only the names without the probability ranges) of the classes above. Think step by step before answering.\n\n

B Details on computing token probabilities

In Section 3, we extract token-level signals through the average probability of tokens after "\n\n". In practice, we store the top-20 logits at each step and restrict our analysis to tokens whose average generation probability exceeds 0.02 and that appear on average more than 20 times per question, ensuring statistical reliability. We merge the probabilities of tokens that share the same semantic context but differ in surface form due to capitalization or leading whitespace (i.e., "Wait", "wait", " Wait", and " wait"). We conduct experiments on 30 questions from AIME24 (OpenAI, 2024), 100 questions from GPQA-D (Rein et al., 2024), and 100 questions from MATH-500 (Lightman et al., 2023).

---

[1] `https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B`

[2] `https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-14B`

[3] `https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B`

[4] `https://huggingface.co/simplescaling/s1.1-32B`

[5] `https://huggingface.co/simplescaling/s1-32B`

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

[7] `https://huggingface.co/datasets/simplescaling/aime24_nofigures`

[8] `https://huggingface.co/datasets/Idavidrein/gpqa`

[9] `https://huggingface.co/datasets/simplescaling/openaimath`

[10] `https://github.com/EleutherAI/lm-evaluation-harness`

[11] `https://github.com/simplescaling/s1`

Model	Per-trace	Group
R1-32B	0.5802^{***}	0.5993^{***}
QwQ-32B	0.7200^{***}	0.6585^{***}
s1.1-32B	0.6896^{***}	0.6092^{***}
s1-32B	0.5312^{***}	0.5019^{**}

Table 8: **Correlation between the correct-incorrect token probability gap and model confidence based on log probability**. "Per-trace" denotes the confidence averaged over a single trajectory, while "Group" refers to the log probability-based confidence following DeepConf (Fu et al., 2025). Pearson correlations are reported (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$).

We also consider calculating token probabilities at different token positions, not only after "\n\n". When token probabilities are computed after a dot or across all positions, the associated tokens for R1-32B in Table 1 remain consistent. However, the strength of the signals is weakened. For example, the average true-associated probability $\bar{p}_{\text{true}}(t)$ of "Wait" is 15.4% when considering positions after "\n\n", but it decreases to 2.15% when considering positions after a dot and to 0.03% when considering all positions. Furthermore, comparatively uninformative tokens, such as "$" or "=", appear to exhibit spurious signals due to problem-specific biases. Since our goal is to focus on discourse markers, we define token-level signals to be computed only at positions following "\n\n".

C Token-level signals and model confidence

In Section 3, we obtain model confidence using self-reported confidence. Model confidence can also be estimated by the average log probability over a single generated trace (per-trace confidence), as used in (Farquhar et al., 2024), or by group confidence, as suggested in (Fu et al., 2025). In this analysis, we follow DeepConf-low and compute group confidence using the bottom 10% local token groups within a trajectory. Table 8 presents the correlation results between our token-probability–based signals, derived from correct- and incorrect-associated tokens, and model confidence measured by log probability. Although the correlation is weaker than that observed with self-reported confidence, which exhibits clearer stratification due to its discrete confidence levels, the correlation remains statistically meaningful.

D Details on token suppression

For token suppression, we apply a masking strategy that prevents the generation of the correct- or incorrect-associated tokens listed in Table 3 during decoding, as a form of token-level steering. Each setting is run with a temperature of 0.6 for three trials, and we report the averaged results.

E The use of LLMs

We use large language models only for minor language refinement, such as improving fluency and clarity. They were not involved in any aspect of the study's design, analysis, or interpretation, and all research findings are entirely our own.