ON THE INTERPLAY BETWEEN BACKTRACKING AND LATENT CONFIDENCE DIRECTIONS IN REASONING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning models deliberate over multiple answer proposals, often resulting in excessively long traces due to backtracking, even when the correct answer has already been reached (On AIME 2026 with Olmo 3 32B Think, ∼87% of rollouts contain 1 unique answer yet still make ∼8 answer proposals on average). Given this potentially "decorative" chain of thought (CoT) reasoning, can we look to the model internals to determine if the model will backtrack or commit to an answer proposal? Operating under the hypothesis that models maintain an internal representation of their uncertainty, we seek to extract this representation and characterize its relationship with backtracking behavior at answer proposals. By querying the model's verbalized confidence following answer proposals throughout a response trace, we extract a linear direction for latent confidence that separates high and low verbalized confidence. This direction can steer verbalized confidence as expected, but it surprisingly can also steer the next token probability of backtracking, performing on par with a linear direction extracted directly from backtracking examples. We then test the direction under progressively more realistic interventions. Steering continuously from an answer proposal modulates response length (∼-1k – ∼+3k tokens) and the number of answer proposals (-4.52 – +3.26) symmetrically in both directions, with minimal cost to accuracy. Applying the confidence direction from the start of generation, where proposal positions are unknown, still shortens traces at comparable accuracy, and its effect surfaces in the linguistic confidence of proposals. Overall, our study reveals the intricate interplay between a model's internal confidence and its proclivity to backtrack or commit to an answer.

1 INTRODUCTION

Reasoning models (LLMs) (Guo et al., 2025; Yang et al., 2025a) solve difficult math (Shao et al., 2024), science, and programming problems (Jiang et al., 2026) by generating long chains of thought (CoT) (Wei et al., 2022) to reason through the problem before arriving at a final answer. A defining characteristic of these traces is backtracking Gandhi et al. (2025), where the model chooses to revise its work during its CoT. Although backtracking behavior can improve performance by helping the model to fix its mistakes and explore alternate possible solutions (Muennighoff et al., 2025; Kim et al., 2025; Yang et al., 2025b; Shen et al., 2025; AI et al., 2025; Tian et al., 2025; Cai et al., 2026, *i.a.*), it also often leads to excessively long or redundant outputs (Chen et al., 2025; Su et al., 2025; Wang et al., 2025).

We study backtracking behavior at *answer proposals*, sentences in which the model arrives at a candidate answer and could plausibly stop reasoning. A model may propose an answer to a question multiple times before committing to a final answer, though we observe that they rarely change their answer value from the initial proposal. Across five benchmarks and three open reasoning models, 84–89% of rollouts propose only a single unique answer, yet the model proposes that answer 4-9 times on average before terminating reasoning (Table 12).

This raises the question: *when a reasoning model proposes an answer, what factors in the internal representations determine whether it commits or backtracks?* From qualitative inspection, a model's

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.

linguistic confidence varies across a reasoning trace, and typically increases steadily as the model nears answer commitment. Thus, a natural hypothesis is that reasoning models maintain an internal representation of confidence in their reasoning progress, which not only affects linguistic confidence but also drives answer commitment. We aim to extract an approximation of this internal confidence and measure its causal role in controlling answer commitment.

Uncertainty quantification (Shorinwa et al., 2025) work demonstrates that LLMs' verbalized confidence (Yoon et al., 2025) is a calibrated confidence measure for answer correctness. We query models' verbalized confidence at each answer proposals in the CoT rollout, a sort of "textual probe", and fit a contrastive linear direction on the model's residual stream representations that separates high from low verbalized confidence. Applying this direction at different points in the CoT not only steers verbalized confidence, but also the model's likelihood of backtracking, indicating that these two tasks share some internal structure in activation space (§4). Interestingly, the cross-task transfer does not hold in the reverse direction: a linear direction that separates and effectively steers backtracking vs. answer commitment probability cannot steer verbalized confidence. Finally, to better understand this confidence direction's causal effect on CoTs, we verify that it both reduces (for positive steering) and increases (for negative steering) the number of answer proposals and overall CoT length as expected (§5.2)– for example, positive steering following mid-trace answer proposals reduces additional answer proposals by 4.52 proposals and response length by 1227 tokens, averaged across three models. When steering the full CoT, linguistic markers of uncertainty change in expected ways (§5.3). Our results provide evidence that latent confidence signals can be 1) extracted by prompting a model to elicit their verbalized confidence and 2) accumulated to approximate a linear confidence direction that can be used to 3) effectively steer answer commitment.

To summarize, our contributions are:

- **A measurement of answer-proposal redundancy.** We define answer proposals, annotate them with a validated LLM-judge pipeline, and show that across three models and five benchmarks most deliberation after the first proposal re-confirms the same answer.
- **Cross-behavior transfer from self-report to control.** A confidence direction extracted from verbalized self-reports causally steers answer commitment, matching a direction fit on commitment itself, while the reverse transfer fails.
- **Trace-level effects on reasoning length.** Confidence steering applied from the start of generation controls the number of answer proposals and response length in both directions.
- **Latent versus surface confidence.** Activation steering is a complementary lever on commitment to editing the linguistic confidence of the chain of thought.

2 Related Work

Interpreting Reasoning and Backtracking A growing body of work studies how reasoning models internally represent and control behaviors such as backtracking. One line of work uses sparse dictionary learning to find unsupervised linear features associated with reasoning patterns and interpret them post-hoc (Venhoff et al., 2026; Zhang et al., 2026; Troitskii et al., 2025). On the other hand, Zhang et al. (2025) and Boppana et al. (2026) show that probes can decode answers or predict answer accuracy before models finish their reasoning trace, and use these probes as a confidence proxy for early exiting. Similarly, Venhoff et al. (2025) extract steering vectors for reasoning behaviors and evaluate them on how well they control the frequency of the behavior they were trained on. Our work differs in two ways. First, rather than fitting a direction on the behavior we aim to steer, we derive it from the model's own verbalized confidence and demonstrate its generalization to a different behavior, answer commitment. Second, we evaluate the confidence direction on various task-level metrics (response length, number of answer proposals, accuracy) across RL-trained and distilled reasoning models.

Verbalized Uncertainty Quantification Uncertainty quantification (UQ) (Geng et al., 2024) is concerned with obtaining a confidence metric that is calibrated with the model's probability of answer correctness. Prompting the model to verbalize its uncertainty (Lin et al., 2022; Tian et al., 2023; Yang et al., 2026) is a widely studied approach. While past research questions the reliability of verbalized uncertainty (Groot & Valdenegro-Toro, 2024; Xiong et al., 2024), compared to token-

Table 1: We pool the rollouts from AIME 2026, GSM8K, OMEGA-500, GPQA-D, MMLU-Pro and calculate the % of rollouts that have n unique answers. For each n, we record the mean $\pm$ std number of proposal markers. Note that std is high because we average over five datasets. The All column computes over all rollouts. See Table 12 for per-dataset results.

Model	Total # rollouts	1 unique		2 unique		3+ unique		All
		% rollouts	proposals	% rollouts	proposals	% rollouts	proposals	proposals
OLMo-3-32B	3202	84.4	8.6 ± 6.5	11.7	16.8 ± 12.2	3.8	26.6 ± 21.6	10.2 ± 9.4
Qwen3-32B	3197	89.3	6.0 ± 4.9	8.0	11.3 ± 7.7	2.7	23.5 ± 16.4	6.9 ± 6.6
R1-Qwen-32B	3025	87.6	4.7 ± 4.3	9.3	11.2 ± 10.6	3.1	16.7 ± 13.4	5.7 ± 6.3

level (Duan et al., 2024) or resampling (Lin et al., 2024) approaches, recent work finds that modern reasoning LLMs' verbalized confidence is well calibrated (Yoon et al., 2025). Kumaran et al. (2026) investigates how models compute verbalized confidence by modeling hidden state interactions with answer tokens, finding that the verbalized confidence is a rich reflection of model self-evaluation beyond token log probabilities. Rather than predicting *final* answer correctness, we leverage verbalized confidence prompting to extract a latent confidence direction and measure its effects across *intermediate* answer proposals.

3 Methodology

3.1 Obtaining Answer Proposals

Answer Proposal Definition A reasoning trace alternates between computation blocks, where the model works towards an answer, and answer proposal sentences. After each intermediate answer proposal, the model typically begins the next computation block expressing its intent to revise or verify its work. While backtracking may occur within computation blocks, we study backtracking vs. commitment behavior immediately following answer proposals, since answer proposals serve as a natural point to break the trace into blocks.

We define a sentence to be an *answer proposal* if it satisfies two conditions: 1) it contains an answer value, and 2) if read up to that point, it is plausible that the model can commit to the answer and end reasoning within the next couple sentences. A sentence is an answer proposal only if the model is arriving at an answer (e.g., "Therefore, I think the answer is *X*."), not simply questioning if the answer is correct (e.g., "Let me make sure *X* is the correct answer.") or hedging a guess (e.g., "Maybe the answer is *X*.").

Extracting Answer Proposals We identify answer proposal positions within the reasoning traces (i.e., within the `<think>` `</think>` tags) across sampled reasoning model generations using an LLM judge pipeline. Compared against golden label human annotations by the authors, the pipeline achieves an average precision of 0.89. After the judge pipeline, we manually review rollouts to ensure that the last answer proposal is always selected. See App. A for additional details. For the rest of the paper, we refer to the last token of an answer proposal sentence as a *marker*. Given n markers in a rollout, we denote the first $n-1$ as mid-trace markers and the nth marker as the final.

We perform this process on sampled reasoning trace rollouts generated by Qwen 3 32B (Yang et al., 2025a), Olmo 3 32B Think (Olmo et al., 2025), and DeepSeek R1 Distill Qwen 32B (DeepSeek-AI, 2025)[1] on the AIME 2026 (Zhang & Math-AI, 2026), GSM8K (Cobbe et al., 2021), OMEGA-500 (Sun et al., 2024), GPQA Diamond (Rein et al., 2023), and MMLU-Pro (Wang et al., 2024) datasets. The first three are mathematical reasoning datasets and the latter two general knowledge multiple-choice questions. We sample 200 prompts from each dataset and sample 4 reasoning traces per prompt from each model with max. response length of 32,768 tokens using a temperature=0.6 and top-p=0.95.[2] Table 1 records the number of unique answers and number of proposal markers per model and dataset. Averaged across the rollouts of all datasets, models propose one unique answer around 90% of the time and make 8 proposals on average, indicating a high propensity to double check and verify their proposed answers.

[1] Subsequently appreviated as "Qwen 3 32B", "Olmo 3 32B", and "Distill 32B", respectively.

[2] We only retain traces where the model reached a final prediction within the max. reasoning budget.

3.2 EXTRACTING LATENT CONFIDENCE

We seek to measure what causes a model to reconsider or commit at an answer proposal. We hypothesize that reasoning models maintain a representation of their confidence surrounding their progress on the problem, and that they typically commit once they are confident enough. As the model works through and double checks its answers, it gradually increases its confidence, thus leading to answer commitment. In this section, we detail how we approximate latent confidence as a linear direction.

Querying Confidence At each selected answer proposal marker, we append the prompt, `My confidence in this answer, on a scale from 0 to 9, is` , and extract the expected confidence from the probabilities of the digit tokens. In this way, we let the model evaluate its confidence in its answer, and verbalize it as an ordinal value. Across all models and datasets, verbalized confidence and answer commitment have a modest pooled Spearman $\rho = 0.15$, and a within-rollout $\rho = 0.22$, as shown in Table 13.[3] We view verbalized confidence as a surface-level approximation of the model's underlying confidence, and ask if we can extract a latent direction from the verbalized confidence data that steers other confidence-related traits.

Extracting confidence direction We divide the answer proposal markers into train, val, and test splits (0.4/0.2/0.4) such that all markers from each trace belong to the same split and the traces within each split differ in accuracy by no more than 5%. To extract the confidence direction, we assign markers as high or low confidence based on the train split's median verbalized confidence value, and obtain the Difference-in-Means (DiM) vector (Rimsky et al., 2024; Belrose, 2023). We extract

$$v^s_{\text{conf}} = \frac{1}{|M^+|} \sum_{p \in M^+} \mathbf{h}^s(p) - \frac{1}{|M^-|} \sum_{q \in M^-} \mathbf{h}^s(q)$$

where M^+, M^- are the high and low confidence markers, and h^s is the site s representation at the last token position of the verbalized confidence prompt. To account for the difference in the number of available traces across different datasets, we sample from the train split so that $|M^+| = |M^-| = 100$. At inference-time, we add or subtract v^s_{conf} at the post-attention or post-MLP residual states. Formally, given a model with N transformer layers, the model has $2N$ steering sites at this layer of granularity; we thus obtain $2N$ DiM directions.

Steering verbalized confidence (VC) For each model, we computed five sets of v_{conf}, one from each dataset's train split. To verify that our confidence directions can steer on the task in which they were fit, we test v_{conf} on steering VC at the last token position of the VC prompt via activation addition (Turner et al., 2024), where positive steering should increase VC and negative steering should decrease VC. We perform a validation sweep across the steering sites with v_{conf} for each dataset. Details on the setup are in App. E. The bottom row in Figure 1 shows the averaged Δ VC across all datasets for $\alpha = \pm 1$ in orange. For context, we also plot the difference between the marker's unsteered VC and the 2nd and 98th percentile VC values, averaged across all markers in the dataset. This headroom represents the approximate upper bound for positive and negative steering, based on the observed VC values that the model naturally emits. As expected, we find that the confidence direction easily steers VC, and it saturates in performance after reaching early-middle layers in Olmo 3 32B, and middle-late layers in Qwen 3 32B and Distill 32B.

Confidence direction robustness The cosine similarity across all datasets' v_{conf} per model is at least 0.85, implying that a general confidence direction is shared across datasets. To ensure that the confidence direction is not encoding a specific numerical difference, we query verbalized confidence using two alternative prompts: `My confidence in this answer, on a scale from 1 to 100, is` and `My confidence in this answer, chosen from "low", "medium", "high", is .` We validate that the confidence direction is robust to the phrasing of the verbalized prompt in App. C. Going forward, we use the confidence direction extracted from the range 9 prompt.

[3] UQ work often also uses token-level entropy as an uncertainty measurement for the model's final answer (Fadeeva et al., 2024; Grünefeld et al., 2026). We measure the sentence-level entropy of mid-trace and final answer proposals and find no significant relationship.

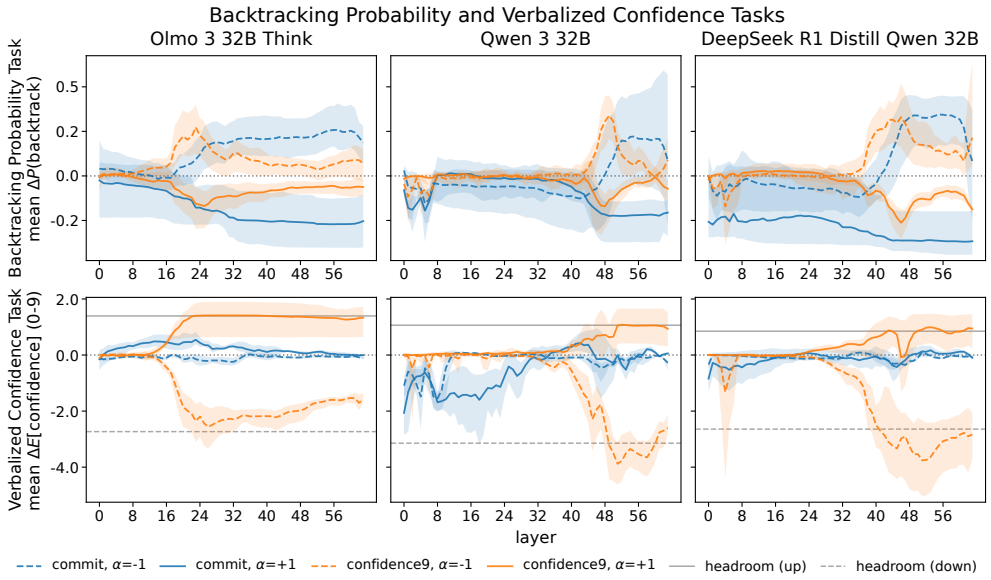


Figure 1: **Top**: Steering a model's probability of emitting a backtracking token as the next token after an answer proposal. **Bottom**: Steering verbalized confidence. Results are averaged across AIME 2026, GSM8K, Omega-500, GPQA-D, and MMLU-Pro. See Figure 4 and Figure 5 for each dataset's validation sweeps.

4 Steering Backtracking Probability with Confidence

4.1 Steering Backtracking Token Probability

As shown in Table 10, a mid-trace answer proposal is followed by a backtracking word ("Wait", "But", "Hmm", etc.) at least 50% of the time across all models, pooled across datasets. On the other hand, a final answer proposal is followed by a backtracking word $\lesssim$2% of the time. Thus, we treat the next token probability of predicting a backtracking token, p_{bt}, as a proxy for answer proposal commitment, and test the confidence representations on steering p_{bt} at the proposal marker. The desired effect of positive steering should be to decrease p_{bt}, since we aim to push the model towards commitment; conversely, negative steering should increase p_{bt}.

As a baseline comparison, we also construct a linear direction for answer commitment from each dataset by taking the DiM between final proposal markers and mid-trace proposal markers. We denote this commitment direction as v_{commit}. Given the difference in the frequency of backtracking words that follow a mid-trace marker and a final marker, v_{commit} should perform well on steering p_{bt}. With v_{commit} and v_{conf}, we apply single token steering at proposal markers with both positive and negative steering. For each dataset's confidence direction, we select the best steering site and α based on its combined performance on positive and negative steering on the validation split (criteria in App. E). Steering performance is generally monotonic so the best $|\alpha| = 1$, and the best layer is consistent within a narrow band of layers across all datasets. We select the best confidence direction and commitment direction across the datasets for p_{bt} and VC steering by testing on a held out test set of markers of 500 markers, constructed by sampling 100 markers from each of the five dataset's test split. We sample the markers to have an even number of final proposal markers and mid-trace markers. Details on datasets and results on test sets are in App. B and E, respectively.

Cross task generalization is asymmetric The top row of Figure 1 plots the validation sweeps for steering Δp_{bt}, averaged across each dataset's direction and validation split, with $\alpha = 1$. The effective steering regions are different between the two directions. The effect of v_{conf} forms a peak in the middle layers, whereas v_{commit} rises in the middle and stays relatively constant through the final layers. v_{conf} peaks in performance earlier than v_{commit} too. Interestingly, the peak in performance for steering Δp_{bt} is synchronized with the initial increase in performance for steering verbalized confi-

Table 2: p_{bt} and VC steering with each model's best v_{commit} and v_{conf} on the marker test set. Gap is the absolute distance between the positive- and negative-steering effects.

		Δp_{bt}					Δ VC				
Model	Vector	Dataset	Site	Negative	Positive	Gap	Dataset	Site	Negative	Positive	Gap
Olmo 3 32B	v_{commit}	GPQA-D	43.attn	+0.258	-0.298	0.556	GSM8K	24.attn	-0.471	+0.609	1.080
	v_{conf}	AIME 2026	23.attn	+0.361	-0.224	0.585	GPQA-D	23.attn	-3.502	+1.215	4.717
Qwen 3 32B	v_{commit}	GPQA-D	51.mlp	+0.399	-0.242	0.641	GSM8K	48.attn	-0.627	+0.595	1.222
	v_{conf}	AIME 2026	47.attn	+0.423	-0.222	0.645	MMLU-Pro	51.attn	-3.284	+1.002	4.286
Distill 32B	v_{commit}	MMLU-Pro	46.attn	+0.432	-0.382	0.814	AIME 2026	48.mlp	-0.290	+0.273	0.563
	v_{conf}	GSM8K	46.attn	+0.403	-0.341	0.744	AIME 2026	51.attn	-5.198	+0.891	6.089

dence, suggesting that the layers that begin to separate high and low verbalized confidence markers also carries a transferrable signal for controlling p_{bt}. Overall, despite not being fit at the marker position, the confidence direction's peak steering effect is comparable to the peak commitment direction, which was fit at the marker positions, thus demonstrating cross-task generalization in the middle band. Table 2 shows the best v_{conf} and v_{commit} for steering p_{bt} and VC.

Since v_{conf} transfers to steering p_{bt}, we check if v_{commit} performs well on steering confidence. As shown in Figure 1, this cross-task generalization does not hold for the commitment direction. In other words, model pathways to emit a backtracking token can be modulated using v_{conf} but the model's verbalized confidence cannot be effectively steered using v_{commit}. Within the 10-layer window centered at v_{conf}'s best site for steering p_{bt}, the cosine similarity between v_{conf} and v_{commit} is 0.028 on average and at most 0.122 across all models.

Regarding steering p_{bt}, as shown in Table 18, each model's best dataset's v_{conf} wins on nearly all datasets from the test set. Even though it was fit on only one dataset, it beats out other datasets' v_{conf} on their own dataset's test markers. This suggests that the confidence direction is generalizable rather than dataset-dependent, though some datasets can elicit the extraction of the confidence direction better than others. In particular, the best confidence directions all come from math datasets. Unless specified otherwise, later experiments use the best confidence and commitment directions based on steering p_{bt}. We also demonstrate in App. D that the confidence direction cannot be trivially explained as merely a next-token probability decoder or a position detector.

Confidence direction decodes meaningfully We take the logit lens of the best v_{conf} per model and find that the top tokens correspond to concepts semantically related to confidence. Logit lens with Olmo 3 32B v_{conf} decodes tokens like 'questionable' and 'controversial' in the negative direction, and 'flawless' and 'effortless' in the positive direction. Qwen 3 32B and Distill 32B decode similar concepts in a multilingual space. Full results of top 10 tokens are in Table 25.

5 Multi Token Steering

Since steering p_{bt} is a limited approximation of answer commitment behavior, we expand our evaluation to more realistic multiple token generation settings. Specifically, we compare our activation-based steering with linguistic interventions (§5.1) and evaluate multi-token steering from an answer proposal (§5.2) and from scratch (§5.3).

5.1 Surface-level confidence versus latent confidence

Models may naturally vary their linguistic confidence or propensity to commit to an answer over the course of the reasoning trace. For example, confident proposals such as `Therefore, I am confident that the answer is X` are more likely to appear at the end of the trace than the middle. Similarly, models may produce consecutive answer proposal sentences when they are on the verge of committing. Thus, is a confidence signal in the surface-level CoT or the model's hidden states more effective at causing the model to commit to its answer?

We filter the 500 marker test set for mid-trace markers whose sampled next token is a backtracking marker. At these mid-trace proposals, we either replace the proposal sentence with a confident

Table 3: Intervening on linguistic confidence versus latent confidence. $\Delta_{\texttt{</think>}}$ and Δp_{bt} are reported as percentages against each model's positive and negative baselines (in Table 23). $\Delta_{\texttt{think}} > 0$ implies quicker termination of reasoning.

Model	Metric	Positive: confident proposal (mid-trace markers)						Negative: hedged proposal (commitment markers)					
		transplant		append		steer (α)		transplant		append		steer (α)	
		neu.	conf.	neu.	conf.	0.5	1	neu.	unc.	neu.	unc.	-0.5	-1
OLMo-3-32B	$\Delta_{\texttt{</think>}}$	$+1.6$	$+3.0$	$+6.2$	$+$**15.9**	$+8.2$	$+10.5$	-4.9	-18.4	-0.8	-19.0	-12.6	$-$**22.1**
	Δp_{bt}	-0.9	$+1.0$	$+6.3$	-2.7	-27.2	$-$**47.9**	$+9.6$	$+30.3$	$+4.2$	$+36.4$	$+22.8$	$+$**40.1**
Qwen3-32B	$\Delta_{\texttt{</think>}}$	$+2.9$	$+6.6$	$+18.2$	$+$**27.3**	$+11.8$	$+15.6$	-3.1	-15.8	$+1.8$	-4.6	-17.1	$-$**27.8**
	Δp_{bt}	$+0.4$	$+0.9$	-3.4	-13.9	-41.3	$-$**70.3**	$+9.5$	$+31.6$	-1.6	$+11.9$	$+30.9$	$+$**55.3**
R1-Qwen-32B	$\Delta_{\texttt{</think>}}$	$+1.8$	$+15.2$	$+23.8$	$+$**44.2**	$+21.9$	$+26.2$	-1.8	-16.7	$+5.1$	-3.7	-13.4	$-$**21.9**
	Δp_{bt}	$+6.5$	-6.2	-10.8	-31.0	-46.3	$-$**70.8**	$+9.1$	$+32.5$	-4.7	$+16.0$	$+29.5$	$+$**40.6**

proposal or append the confident proposal. We then measure the change in the next token probability of backtracking and if the model generates </think> within the next 128 tokens. We compare this against steering the last token of the answer proposal sentence. In this manner, we measure how much the answer proposal's surface-level confidence interacts with the internal confidence of the model. We generate 4 rollouts per marker to get a more accurate measure of $\Delta_{\texttt{</think>}}$.

We test the reverse setting as well: At final proposal markers, we filter for commitment markers that are within 12 tokens from the </think> token. We then either replace or append that proposal sentence with an uncertain sentence containing the answer value, and measure the increase in next token backtracking probability and the fraction of responses that end reasoning within the next 256 tokens. We choose 256 instead of 128 to allow the model some buffer for possible wordy "superficial" backtracking. We apply the same steering intervention with negative α.

We employ three linguistically confident answer proposals and three linguistically uncertain sentences hedging the answer proposal. We also use a neutral sentence to control for the variation resulting from manipulating the CoT out of the model's natural distribution. The sentences are listed in Table 22. Results are shown Table 3, with the results of the three confident/uncertain sentences averaged to a single column.

Answer proposal replacement has a modest effect that is similar to or slightly worse than steering with $\alpha = 0.5$ across both settings. The effects of answer proposal addition is asymmetric. Although it is most effective at answer commitment in the positive setting, it is unreliable in the negative setting for Qwen 3 32B and Distill 32B, performing worse than proposal replacement. Steering with $|\alpha| = 1$ outperforms both CoT manipulation methods in the negative setting, and it trails behind answer proposal addition in the positive setting.

Overall, we find that adding latent confidence only to the last token of the answer proposal sentence is more reliable at modulating answer proposal backtracking compared to CoT confidence manipulation.

5.2 Confidence steering controls reasoning length

§4.1 and §5.1 demonstrate that the confidence direction controls p_{bt} and $\Delta_{\texttt{</think>}}$ through single token steering, but how does a consistent multi-token steering presence affect the rest of the model's generations? To answer this question, we continue generation from an answer proposal while steering at each generated token to measure how much a continued steering presence on newly generated tokens affects the model's behavior toward answer commitment and backtracking.

We continue steering the model after the last answer proposal token to measure if the direction reduces the number of answer proposals while maintaining accuracy. We sample using temp=0.6 and topp=0.95 generations after each proposal, and sweep across alpha values (0.1, 0.25, 0.5, 0.75) and (-0.1, -0.25, -0.5). The max total response length is capped at 32768.

We measure the change in the remaining response length, number of additional answer proposals, and overall accuracy. We report separate results for mid-trace and final markers since final proposals are already near reasoning termination. As shown in Figure 2, v_{conf} changes commitment over

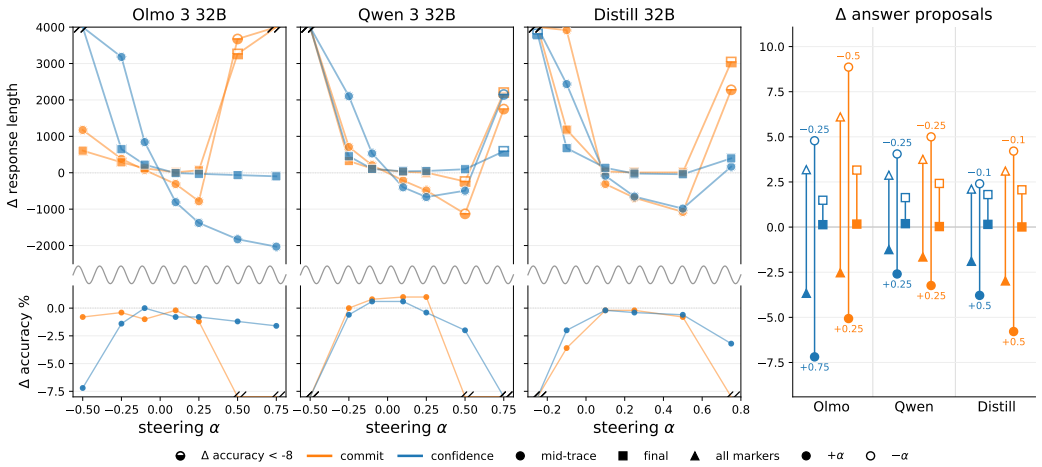


Figure 2: Post Marker Continuation Steering Length and accuracy shift vs α. With $\alpha > 0$, negative Δ response length is better. With $\alpha < 0$, positive Δ response length is better. We see that confidence steering matches or outperforms steering with the commitment direction, when constrained to keep Δ accuracy close to 0.

Table 4: Steering with the confidence direction from the start of the rollout reduces response length from 400-2000 tokens on average, while maintaining comparable accuracy. Delaying the steering by 5000 tokens helps maintain accuracy, with minimal implementation cost and has no impact on inference speed.

Model	Arm	AIME 2026		GSM8K		OMEGA-500		GPQA-D		MMLU-Pro		Avg Δ	
		Acc	Len	Acc	Len	Acc	Len	Acc	Len	Acc	Len	ΔAcc	ΔLen
Olmo 3 32B	Unsteered	80.8	15788	95.4	1939	61.9	16237	61.5	12686	71.9	6086	–	–
	$\alpha = 0.1$	75.8	14328	94.8	1512	59.2	14399	58.8	10966	71.1	4864	-2.3	-15.2%
	$\alpha = 0.1$, d=5000	79.2	14850	95.2	1912	59.9	14835	62.1	11357	71.4	5547	-0.7	-7.1%
Qwen 3 32B	Unsteered	71.7	14537	94.5	1820	53.5	14857	67.0	7781	76.0	3904	–	–
	$\alpha = 0.1$	70.0	13580	93.8	1494	54.4	14070	67.4	7299	75.0	3660	-0.4	-8.4%
	$\alpha = 0.1$, d=5000	70.8	13428	94.4	1844	55.4	13917	67.7	7713	76.2	3850	+0.4	-3.0%
Distill 32B	Unsteered	63.3	12088	94.6	1502	45.6	12878	60.5	6753	70.2	3717	–	–
	$\alpha = 0.1$	60.8	9846	95.0	1112	40.8	10533	59.2	5471	72.5	3028	-1.2	-20.0%
	$\alpha = 0.1$, d=5000	58.3	10831	94.9	1372	44.8	11030	62.2	6145	70.6	3368	-0.7	-10.4%

the whole trace in both directions, comparable to the commitment direction. With appropriate α, positive steering with v_{conf} reduces the number of answer proposals by 7.19, 2.60, and 3.78 on Olmo 3 32B, Qwen 3 32B, and Distill 32B, respectively, as well as the response length at little cost to accuracy. Negative α steering conversely increases the number of answer proposals by 4.78, 4.05, and 2.40, respectively, as well as response length. Steering final proposal markers with $+\alpha$ slightly increases response length on Qwen 3 32B, though this is largely sampling noise, since the model is already close to ending generation. Raw results are in Table 24.

5.3 Steering from Scratch

Steering the model from an answer proposal marker can reliably modulate verbalized confidence and the number of answer proposals. In particular, positive steering after the answer proposal position with the latent confidence direction can reduce token count.

We extend this to steering at inference, where the answer proposals are unknown by measuring how to steer the model generations from scratch as well. We steer with a small $\alpha = 0.1$ to avoid degenerate rollouts. Since reasoning models typically spend the initial part of the reasoning trace exploring and formulating plans, with answer proposals arising later into the trace, we test a second setting where we delay steering until $d = 5000$ tokens have been generated to allow the model to first reason without intervention.

Results are shown in Table 4. Steering with small $\alpha = 0.1$ reduces response length without a significant decrease in accuracy. As we do not optimize for the best α or use custom delays for various datasets, we present these results not as a practical efficiency method but as a proof of concept that the effect of confidence steering holds without knowing the exact positions of the proposals in advance. We use vLLM (Kwon et al., 2023) for all steered rollouts.

Steering answer proposal linguistics Lastly, having established that the latent confidence direction modulates trace-level statistics of response length and answer proposal count, we qualitatively inspect if steering with the latent confidence direction affects the linguistic nature of the answer proposal sentence itself. At answer proposals markers, we regenerate the answer proposal sentence with positive and negative confidence steering at $\alpha = \pm 0.5$ and continue steering up to 128 tokens. With positive steering, models typically regenerate their original proposal, or reword more confidently (see Table 19). Similarly, with negative steering, regenerated answer proposals express uncertainty (Table 21) or abandon the proposal entirely in favor of backtracking immediately (Table 20).

Conclusion and Limitations

To better understand the relationship between model confidence and backtracking, we provide evidence that latent confidence signals can be extracted through verbalized confidence elicitation and accumulated to approximate a linear confidence direction that can be used to steer answer commitment. Our results reveal that the pathway for verbalized confidence in reasoning models also sits upstream for controlling answer proposal backtracking and commitment, demonstrating that latent confidence signals are entangled with backtracking behavior. We also provide comparisons between and interactions with surface-level confidence and latent confidence, finding that 1) single token steering elicits stronger effects than increasing the linguistic confidence of the answer proposal, and 2) the effects of confidence steering are reflected in the model's linguistic confidence.

Since model internal confidence is likely a multi-faceted representation that cannot be captured linearly, we do not claim that model confidence is fully explained by our constructed direction. Rather, we defend that the linear direction separating the model's verbalized assessment of its confidence transfers effectively towards steering behavior at answer proposals. We notice that un-steered reasoning traces occasionally terminate with a resignation phrase like `Given how much time I have spent, I think I will go with X`. In these cases, the model is not verbally confident, but it commits anyways, demonstrating that high confidence may be a sufficient but not a necessary criteria for answer commitment. An avenue for future work is to measure how the linear confidence direction evolves across different reasoning post training strategies, as well as to explore nonlinear confidence directions.

AI use statement

In this work, we used generative AI tools to design or provide feedback on research methodology or experiments, implement methods, clean and reformat datasets, support qualitative and thematic data analysis, and interpret results. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, or assist with translation. Additionally, we used generative AI tools to assist with plot and figure generation and to identify relevant related work. The rest of the disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

All models and datasets used in this work are publicly available, and we report the sampling parameters (temperature, top-p, maximum response length) and prompt subsampling used to generate rollouts in §3.1. The LLM judge pipeline for identifying answer proposals is described in App. A, including the verbatim prompt templates for both stages (Table 6, Table 7), the manual review procedure, and agreement against author annotations (Table 5). §3.2 specifies the verbalized con-

fidence prompt, the Difference-in-Means construction, and the train/validation/test split procedure. Per-model and per-dataset marker counts are in Table 9 and Table 11. App. E gives the steering implementation, the validation sweep, and the exact best-site selection criterion. We use vLLM for all steered rollouts. We intend to release code for direction extraction, steering, and evaluation upon publication.

REFERENCES

Essential AI, Darsh J Shah, Peter Rushton, Somanshu Singla, Mohit Parmar, Kurt Smith, Yash Vanjani, Ashish Vaswani, Adarsh Chaluvaraju, Andrew Hojel, et al. Rethinking reflection in pre-training, 2025. URL https://arxiv.org/abs/2504.04022.

Nora Belrose. Diff-in-means concept editing is worst-case optimal: Explaining a result by sam marks and max tegmark, 2023. URL https://blog.eleuther.ai/diff-in-means/.

Siddharth Boppana, Annabel Ma, Max Loeffler, Raphael Sarfati, Eric Bigelow, Atticus Geiger, Owen Lewis, and Jack Merullo. Reasoning theater: Disentangling model beliefs from chain-of-thought, 2026. URL https://arxiv.org/abs/2603.05488.

Hongyi James Cai, Junlin Wang, Xiaoyin Chen, and Bhuwan Dhingra. How much backtracking is enough? exploring the interplay of SFT and RL in enhancing LLM reasoning. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=bpVWLDwlu4.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? on the overthinking of long reasoning models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=MSbU3L7V00.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models, 2024. URL https://arxiv.org/abs/2307.01379.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. Fact-checking the output of large language models via token-level uncertainty quantification. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9367–9385, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.558. URL https://aclanthology.org/2024.findings-acl.558/.

Kanishk Gandhi, Ayush K Chakravarthy, Anikait Singh, Nathan Lile, and Noah Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective STars. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=QGJ9ttXLTy.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. A survey of confidence estimation and calibration in large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6577–6595, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.366. URL https://aclanthology.org/2024.naacl-long.366/.

Tobias Groot and Matias Valdenegro-Toro. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models, 2024. URL https://arxiv.org/abs/2405.02917.

Nils Grünefeld, Bertram Højer, Philipp Mondorf, Barbara Plank, Anna Rogers, Christian Hardmeier, Stefan Heinrich, and Jes Frellsen. Tracing uncertainty in language model "reasoning", 2026. URL https://arxiv.org/abs/2605.07776.

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, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. A survey on large language models for code generation. *ACM Transactions on Software Engineering and Methodology*, 35 (2):1–72, January 2026. ISSN 1557-7392. doi: 10.1145/3747588. URL http://dx.doi.org/10.1145/3747588.

Joongwon Kim, Anirudh Goyal, Liang Tan, Hannaneh Hajishirzi, Srinivasan Iyer, and Tianlu Wang. Astro: Teaching language models to reason by reflecting and backtracking in-context. 2025. URL https://arxiv.org/abs/2507.00417.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited, 2019. URL https://arxiv.org/abs/1905.00414.

Dharshan Kumaran, Arthur Conmy, Federico Barbero, Simon Osindero, Viorica Patraucean, and Petar Veličković. How do llms compute verbal confidence, 2026. URL https://arxiv.org/abs/2603.17839.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words, 2022. URL https://arxiv.org/abs/2205.14334.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models, 2024. URL https://arxiv.org/abs/2305.19187.

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 Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Maohao Shen, Guangtao Zeng, Zhenting Qi, Zhang-Wei Hong, Zhenfang Chen, Wei Lu, Gregory W. Wornell, Subhro Das, David Daniel Cox, and Chuang Gan. Satori: Reinforcement learning with chain-of-action-thought enhances LLM reasoning via autoregressive search. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=j4FXxMiDjL.

Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z. Ren, and Anirudha Majumdar. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions, 2025. URL https://arxiv.org/abs/2412.05563.

Jinyan Su, Jennifer Healey, Preslav Nakov, and Claire Cardie. Between underthinking and overthinking: An empirical study of reasoning length and correctness in llms, 2025. URL https://arxiv.org/abs/2505.00127.

Yiyou Sun, Shawn Hu, Georgia Zhou, Ken Zheng, Hannaneh Hajishirzi, Nouha Dziri, and Dawn Song. Omega: Can llms reason outside the box in math? evaluating exploratory, compositional, and transformative generalization. *arXiv preprint arXiv:2506.18880*, 2024.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330/.

Xiaoyu Tian, Sitong Zhao, Haotian Wang, Shuaiting Chen, Yunjie Ji, Yiping Peng, Han Zhao, and Xiangang Li. Think twice: Enhancing llm reasoning by scaling multi-round test-time thinking. 2025. URL https://arxiv.org/abs/2503.19855.

Dmitrii Troitskii, Koyena Pal, Chris Wendler, and Callum Stuart McDougall. Internal states before wait modulate reasoning patterns. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 18640–18649, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1012. URL https://aclanthology.org/2025.findings-emnlp.1012/.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2024. URL https://arxiv.org/abs/2308.10248.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors, 2025. URL https://arxiv.org/abs/2506.18167.

Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Base models know how to reason, thinking models learn when, 2026. URL https://arxiv.org/abs/2510.07364.

Chenlong Wang, Yuanning Feng, Dongping Chen, Zhaoyang Chu, Ranjay Krishna, and Tianyi Zhou. Wait, We Don't Need to "Wait"! Removing Thinking Tokens Improves Reasoning Efficiency, June 2025. URL http://arxiv.org/abs/2506.08343. arXiv:2506.08343 [cs].

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 23650–23678, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/6733cf15e10e2cd1d59af033c3bb8507-Paper-Conference.pdf.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. On verbalized confidence scores for llms, 2026. URL https://arxiv.org/abs/2412.14737.

Shiming Yang, Yuxuan Tong, Xinyao Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=OLodUbcWjB.

Dongkeun Yoon, Seungone Kim, Sohee Yang, Sunkyoung Kim, Soyeon Kim, Yongil Kim, Eunbi Choi, Yireun Kim, and Minjoon Seo. Reasoning models better express their confidence. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=rbBtoVnduo.

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. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=O6I0Av7683.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2026, 2026.

Zhenyu Zhang, Shujian Zhang, John Wheatley Lambert, Wenxuan Zhou, Zhangyang Wang, Mingqing Chen, Andrew Hard, Rajiv Mathews, and Lun Wang. Fantastic reasoning behaviors and where to find them: Unsupervised discovery of the reasoning process. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=e0wfMUiZnY.

A LLM JUDGE PIPELINE

Motivation for Judge Pipeline Extracting answer proposals from a reasoning trace requires scanning rollouts that are thousands of tokens long, which is impractical for humans to do. Furthermore, relying solely on regex based on the model's final prediction would ignore any intermediate answer values the model had proposed and later revised. A similarly simple option is to find all sentences in the trace with the word "answer". However, answer proposal sentences can vary in lexical style, and a sentence does not necessarily need to have the word "answer" for the model to commit at that sentence. An LLM judge can handle the majority of the labeling at scale, with human annotation as a final pass at the end. Since we aim to study representations at answer proposal markers, the LLM judge pipeline should high precision. In other words, for any answer sentence the LLM judge selects to be an answer proposal, the human should have also selected it to be an answer proposal. Additionally, high recall on the final answer proposal is critical, since only one proposal per rollout is the final answer proposal, and it is used to create the commit class for our Difference-in-Means vector.

High recall for mid-trace answer proposals is a lower priority, as there typically many mid-trace answer proposals within a reasoning trace and only one final answer proposal.

We use a two step judge pipeline:

1. For each rollout, we ask the judge to return all unique answer values proposed by the model, including proposed values that were later rejected or revised. We skip rollouts that do not have a final prediction.
2. For each rollout, we tag each sentence in the reasoning trace with a [[Sk]] marker if it contains any answer value returned from step 1, where k is the index id that distinguishes the candidate markers. This filters down the number of sentences the judge has to evaluate. Then, we pass the whole rollout to the judge and ask the judge to select the markers corresponding to the answer proposal sentences.

For MCQA, we skip step 1 and instead use regex to tag all sentences that contain a standalone capital letter from "A"-"D" for GPQA-D and "A"-"J" for MMLU-Pro. The regex rule ensures that the letter does not get selected if it has adjacent letters, digits, or the punctuation [-, ', _, ^, \, =]. Thus (C), C. and option C do match. For letters "A" and "I", which are english words, If the letter is followed by a space and a lowercase word (or, for "A", a number), it is treated as the article or pronoun and skipped ("A benzene ring", "I think"). If an option cue comes before it, such as "option", "choice", "answer" or "letter", we do select it.

Table 6 shows the prompt for step 1. Table 7 and Table 8 show the prompts for step 2 for math and MCQA, respectively. Finally, after the judge pipeline, we manually check each rollouts whose last candidate marker was not selected by the judge, and select the candidate marker if it is indeed an answer proposal. If the last sentence is a hedging statement, such as "So, maybe the correct answer is not listed, but among the options, option (A) is the closest." we do select it, because the model commits with this proposal. If two consecutive sentences both arrive at an answer to the question, i.e. "So the answer is 10. Therefore, I believe the answer is 10", we treat both sentences as answer proposals. This is because it is plausible that the model commits after generating the first answer proposal sentence. Typically, this pattern occurs near the reasoning termination.

Manual check against golden labels We manually annotate answer proposals on 30 rollouts from Olmo 3 32B on AIME 2026, for a total of 332 selected markers, and 30 rollouts from Olmo 3 32B on GPQA-D, for a total of 209 selected markers. We treat these as golden labels, and use them to iteratively design the judge prompt templates for math questions and MCQA questions for all models. Precision and Recall of the final judge pipeline (without the manual check) are shown in Table 5.

Question Format Type	Precision	Recall
Math	0.9129	0.7892
MCQA	0.8676	0.9091

Table 5: Human-LLM Agreement for our Judge Pipeline.

Table 6: Prompt given to the judge model to extract the set of unique proposed answers from a reasoning trace on math tasks. {question} and {response} are replaced with the problem statement and the model's full rollout.

```
You are an expert at parsing language model thinking traces.

Given a question and a language model's response, return a list of all unique solutions
proposed throughout the response.  Do not treat any intermediate values as solutions.  If
the same answer is formatted in multiple ways, then include each format in the list.
Copy each answer verbatim.  Below are examples of answers formatted in multiple ways:

  - "1/2", "\frac{1}{2}", "\dfrac{1}{2}"
- "5" and "\boxed{5}"
- "6+9i" and "6 + 9i"

Here is the question:

<question>
{question}
</question>

Here is the response:

<response>
{response}
</response>

Do not solve the problem yourself.  List all unique solutions proposed throughout the
response.  Do not treat any intermediate values as a solution.  For example, if the
question asks for a fraction expressed as the sum of the numerator and denominator, do
not treat the fraction as an answer.  If an answer is formatted in multiple ways, then
include each format in the list.  Copy each answer verbatim.

Return the list of answers in a JSON: {"answers":  [<str>, <str>, ...]}.
```

Table 7: Prompt given to the judge model to select, from a set of inserted sentence markers, those at which the model commits to an answer. {question} and {response} are replaced with the problem statement and the marker-annotated rollout.

```
You are an expert at parsing language model thinking traces.

You will be shown a question and a language model's response.  A tag [[Sk]] has been
inserted at the END of certain sentences, where k is the marker id.  The tags are
annotations and were not produced by the model.

For each tagged marker, decide whether the sentence BEFORE the marker is one in which the
model commits to an answer to the prompt.  Select only those markers.  Do NOT select a
marker whose preceding sentence is an intermediate step or does not settle on an answer
to the question.

Here is the question:

<prompt>
{question}
</prompt>

Here is the model's response:

<response>
{response}
</response>

A tag [[Sk]] has been inserted at the END of certain sentences, where k is the marker id.
The tags are annotations and were not produced by the model.

First, identify precisely which quantity the problem asks the language model to find.
For each tagged marker, decide whether the sentence BEFORE the marker is one in which the
model commits to an answer for that exact quantity.  Select only those markers.

Do NOT select a marker if:

  - the preceding sentence is an intermediate step or auxiliary quantity that does not
settle on an answer to the question.
  - the preceding sentence ends with a question mark or has an explicitly unconfident
clause, such as "but let me make sure".  It is fine if the sentence AFTER the tag is
uncertain.
  - the preceding sentence contains no English words - it is only symbols, numbers and
operators.  "180 + 24 = 204." and "m+n=100+13=113." are NOT proposals.  A brief
computation phrased as the answer, such as "100 and 13 are coprime, so m=100, n=13, so
m+n=113 is the answer" DOES count.
  - the value is only hedged as a possibility, such as "maybe the answer is 5" or "perhaps
it's 5" - UNLESS the sentence also decides to go with it, as in "Maybe the answer is 113,
so I'll go with that", which DOES count.  A neutral hypothetical such as "the total would
be $90" DOES count.

If you are unsure about a marker, it is better not select it.  The final tagged markers
are the most important to evaluate.  For each selected marker, record its id and the
model's answer.

Return JSON: {"answers":  [{"marker_id":  <int>, "answer":  <string>}, ...]}.  If no
markers qualify, return an empty list.
```

Table 8: Prompt given to the judge model to select, from a set of inserted sentence markers, those at which the model commits to an answer choice on multiple-choice tasks. {question} and {response} are replaced with the problem statement and the marker-annotated rollout.

```
You are an expert at parsing language model thinking traces.

You will be shown a multiple choice question and a language model's response.  A tag
[[Sk]] has been inserted at the END of certain sentences, where k is the marker id.  The
tags are annotations and were not produced by the model.

For each tagged marker, decide whether the sentence BEFORE the marker is one in which the
model commits to an answer choice to the question.  Select only those markers.  Do NOT
select a marker whose preceding sentence is an intermediate step or does not settle on an
answer to the question.

Here is the question:

<prompt>
{question}
</prompt>

Here is the model's response:

<response>
{response}
</response>

A tag [[Sk]] has been inserted at the END of certain sentences, where k is the marker id.
The tags are annotations and were not produced by the model.

First, identify precisely what the problem asks the language model to find.  For each
tagged marker, decide whether the sentence BEFORE the marker is one in which the model
commits to an answer choice.  Select only those markers.

Do NOT select a marker if:

  - the preceding sentence is an intermediate step or auxiliary quantity that does not
settle on an answer to the question.
  - the preceding sentence ends with a question mark or has an explicitly unconfident
clause, such as "but let me make sure".  It is fine if the sentence AFTER the tag is
uncertain.
  - the preceding sentence does not contain an answer choice.
  - the value is only hedged as a possibility, such as "maybe the answer is C" or "perhaps
it's C" - UNLESS the sentence also decides to go with it, as in "Maybe the answer is C,
so I'll go with that", which DOES count.  A neutral hypothetical such as "the total would
be A" DOES count.

If you are unsure about a marker, it is better not select it.  The final tagged markers
are the most important to evaluate.  For each selected marker, record its id and the
model's answer.

Return JSON: {"answers":  [{"marker_id":  <int>, "answer":  <string>}, ...]}.  If no
markers qualify, return an empty list.
```

B Dataset Details

Marker Splits Table 9 shows the total number of rollouts and markers for each model and dataset, which is used to form the train, validation, and test splits to construct the confidence direction and commitment direction. Table 11 details the marker test sets for each model. 100 markers from each dataset's test split are sampled to form the test set.

Rollout Statistics Table 10 shows backtracking frequency per proposal marker type on each model and dataset. Table 12 shows unique answers vs. answer proposals per rollout per model and dataset.

Table 9: Marker Splits for each Dataset and Model after passing the model through the judge pipeline. The number in parentheses indicates the number of prompts used to generate rollouts for each dataset. We subsample larger datasets randomly to have 200 prompts.

	AIME 2026 (30)		GSM8K (200)		OMEGA-500 (200)		GPQA-D (198)		MMLU-Pro (200)	
Model	# Rollouts	# Markers	# Rollouts	# Markers	# Rollouts	# Markers	# Rollouts	# Markers	# Rollouts	# Markers
Olmo 3 32B	113	1175	788	7813	728	7922	787	6511	786	9276
Qwen 3 32B	111	985	791	6099	756	6297	779	3647	786	4898
R1 Qwen 32B	113	600	790	5530	693	5310	709	2674	720	3118

Table 10: Backtracking Next Token Frequency on sample rollouts from each dataset, split by mid-trace proposals and final proposals. We define the following words as backtracking words: `"Wait", But", "However", "Hmm", "Alternatively", "Maybe", "Another"`

Model	Marker type	AIME 2026	GSM8K	OMEGA-500	GPQA-D	MMLU-Pro	All
Olmo 3 32B	Final	3.5%	0.5%	5.2%	0.3%	0.4%	1.6%
	Mid-trace	76.9%	40.6%	68.6%	65.1%	65.2%	60.6%
Qwen 3 32B	Final	3.6%	0.6%	5.6%	0.3%	0.5%	0.9%
	Mid-trace	66.8%	37.8%	63.0%	64.2%	55.6%	52.2%
Distill 32B	Final	4.4%	0.3%	2.5%	2.3%	2.2%	1.9%
	Mid-trace	77.8%	39.6%	71.6%	78.8%	69.9%	61.8%

Table 11: Test Marker Splits Model. # M denotes number of markers. # denotes corresponding number of unique rollouts that contained those markers

		AIME 2026		GSM8K		OMEGA-500		GPQA-D		MMLU-Pro	
Model	Marker type	# R	# M	# R	# M	# R	# M	# R	# M	# R	# M
Olmo 3 32B	Final	41	41	50	50	50	50	50	50	50	50
	Mid-trace	28	59	48	50	45	50	43	50	45	50
	All	41	100	92	100	88	100	85	100	90	100
Qwen 3 32B	Final	42	42	50	50	50	50	50	50	50	50
	Mid-trace	27	58	44	50	30	50	40	50	45	50
	All	42	100	90	100	59	100	84	100	89	100
R1 Qwen 32B	Final	41	41	50	50	50	50	50	50	50	50
	Mid-trace	28	59	41	50	40	50	43	50	40	50
	All	41	100	86	100	86	100	83	100	85	100

Table 12: Unique answers vs. answer proposals per rollout per model and dataset. # unique is the number of distinct answers the judge confirmed in a rollout's reasoning trace; proposals is the mean $\pm$ std of all confirmed answer proposals, repeats included, over the n rollouts in that row.

Model	Dataset	# unique	n	proposals
OLMo-3-32B	AIME 2026	1	107	9.7 ± 5.5
		2	5	23.6 ± 8.3
		3+	1	21.0 ± 0.0
	GSM8K	1	700	8.6 ± 4.6
		2	70	18.6 ± 14.8
		3+	18	26.7 ± 16.2
	OMEGA-500	1	643	9.9 ± 6.7
		2	55	17.4 ± 13.2
		3+	30	19.1 ± 16.6
	GPQA-D	1	620	6.7 ± 5.1
		2	136	13.2 ± 9.5
		3+	31	19.1 ± 13.6
	MMLU-Pro	1	633	8.8 ± 8.6
		2	110	19.3 ± 11.9
		3+	43	37.4 ± 26.6
Qwen3-32B	AIME 2026	1	107	8.7 ± 5.4
		2	3	13.3 ± 6.8
		3+	1	13.0 ± 0.0
	GSM8K	1	712	7.0 ± 3.9
		2	67	14.5 ± 8.3
		3+	12	14.8 ± 5.2
	OMEGA-500	1	644	8.1 ± 6.9
		2	59	12.4 ± 8.9
		3+	27	13.5 ± 7.4
	GPQA-D	1	710	4.1 ± 3.1
		2	57	8.4 ± 6.0
		3+	12	19.2 ± 10.5
	MMLU-Pro	1	681	4.3 ± 3.4
		2	70	9.7 ± 5.6
		3+	35	36.1 ± 17.5
R1-Qwen-32B	AIME 2026	1	97	4.7 ± 3.7
		2	13	7.8 ± 4.5
		3+	3	15.7 ± 6.6
	GSM8K	1	716	6.2 ± 4.1
		2	58	14.8 ± 13.1
		3+	16	12.6 ± 10.0
	OMEGA-500	1	577	6.2 ± 5.8
		2	78	14.4 ± 13.9
		3+	38	16.0 ± 11.7
	GPQA-D	1	627	3.2 ± 2.5
		2	69	7.6 ± 5.4
		3+	13	11.4 ± 7.7
	MMLU-Pro	1	634	3.2 ± 2.9
		2	62	8.7 ± 4.9
		3+	24	23.5 ± 17.7

Table 13: Spearman ρ between verbalized confidence (expected value of the 0 to 9 at each proposed-answer marker) and commitment (1 if the marker is the rollout's final answer, 0 if the model keeps reasoning). *Pooled*: every marker as a point; rows spanning several datasets rank confidence within each dataset first. *Within-rollout*: confidence centered on its rollout's mean, single-marker rollouts dropped. * denotes $p < 0.05$

				Pooled ρ			Within-rollout ρ		
Model	Dataset	Rollouts	Markers	All	Correct	Incorrect	All	Correct	Incorrect
Olmo 3 32B	AIME 2026	113	1175	$+0.21^*$	$+0.25^*$	$+0.07$	$+0.30^*$	$+0.31^*$	$+0.23^*$
	OMEGA	728	7922	$+0.11^*$	$+0.16^*$	$+0.01$	$+0.22^*$	$+0.26^*$	$+0.14^*$
	GSM8K	788	7813	$+0.28^*$	$+0.28^*$	$+0.20^*$	$+0.25^*$	$+0.26^*$	$+0.22^*$
	GPQA-D	787	6511	$+0.02^*$	$+0.06^*$	-0.02	$+0.16^*$	$+0.19^*$	$+0.09^*$
	MMLU-Pro	786	9276	$+0.10^*$	$+0.15^*$	-0.03	$+0.14^*$	$+0.18^*$	$+0.04^*$
	All	3202	32697	$\mathbf{+0.13^*}$	$\mathbf{+0.18^*}$	$\mathbf{+0.00}$	$\mathbf{+0.20^*}$	$\mathbf{+0.23^*}$	$\mathbf{+0.10^*}$
Qwen 3 32B	AIME 2026	111	985	$+0.22^*$	$+0.32^*$	$+0.02$	$+0.32^*$	$+0.33^*$	$+0.28^*$
	OMEGA	180	1626	$+0.14^*$	$+0.25^*$	-0.00	$+0.27^*$	$+0.29^*$	$+0.24^*$
	GSM8K	791	6099	$+0.31^*$	$+0.31^*$	$+0.27^*$	$+0.26^*$	$+0.26^*$	$+0.26^*$
	GPQA-D	779	3647	$+0.04^*$	$+0.11^*$	-0.02	$+0.35^*$	$+0.36^*$	$+0.31^*$
	MMLU-Pro	786	4898	$+0.16^*$	$+0.17^*$	$+0.03$	$+0.22^*$	$+0.23^*$	$+0.21^*$
	All	2647	17255	$\mathbf{+0.18^*}$	$\mathbf{+0.22^*}$	$\mathbf{+0.03}$	$\mathbf{+0.27^*}$	$\mathbf{+0.27^*}$	$\mathbf{+0.25^*}$
Distill 32B	AIME 2026	113	600	$+0.17^*$	$+0.29^*$	$+0.01$	$+0.23^*$	$+0.40^*$	-0.05
	OMEGA	693	5310	$+0.12^*$	$+0.24^*$	$+0.01$	$+0.22^*$	$+0.28^*$	$+0.15^*$
	GSM8K	790	5530	$+0.35^*$	$+0.35^*$	$+0.33^*$	$+0.39^*$	$+0.41^*$	$+0.22^*$
	GPQA-D	709	2674	$+0.01$	$+0.06^*$	-0.04	$+0.19^*$	$+0.29^*$	$+0.01$
	MMLU-Pro	720	3118	$+0.14^*$	$+0.17^*$	-0.09^*	$+0.19^*$	$+0.33^*$	-0.04
	All	3025	17232	$\mathbf{+0.17^*}$	$\mathbf{+0.23^*}$	$\mathbf{+0.00}$	$\mathbf{+0.25^*}$	$\mathbf{+0.33^*}$	$\mathbf{+0.08^*}$
All models		8874	67184	$\mathbf{+0.15^*}$	$\mathbf{+0.21^*}$	$\mathbf{+0.01}$	$\mathbf{+0.22^*}$	$\mathbf{+0.26^*}$	$\mathbf{+0.13^*}$

C Vector Consistency

Verbalized Confidence Prompts We test the robustness of the confidence extraction method to various verbalized confidence prompts on AIME 2026. We use the following prompts:

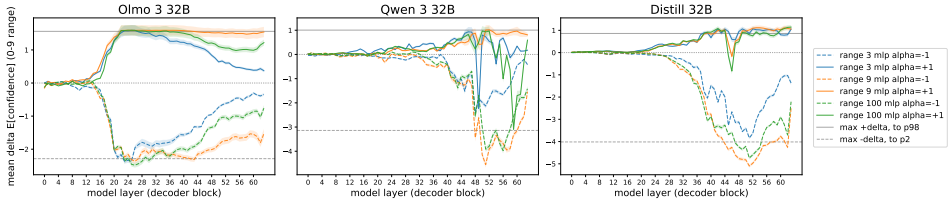
- Range 9: My confidence in this answer, on a scale from 0 to 9, is
- Range 100: My confidence in this answer, on a scale from 1 to 100, is
- Range 3: My confidence in this answer, chosen from "low", "medium", "high", is "

We compare the spearman rank correlation between the rankings of the markers from low to high verbalized confidence per pair of prompts. Across all three models, the rank correlation averaged across the three pairs of prompts is at least 0.932. This demonstrates that each answer proposal marker ranks similarly in its verbalized confidence across the three prompts. Similarly, we take the centered kernel alignment Kornblith et al. (2019) similarity between the representations of the final token extracted between each pair of prompts. While CKA is typically used to measure the similarities of representations between two models while keeping the input the same, an equivalent setup is to measure the similarities of representations between two different prompts while keeping the model and the prefix tokens (prompt and model response) the same. Across all models, the CKA averaged across the three pars of prompts is at least 0.876. See Table 14.

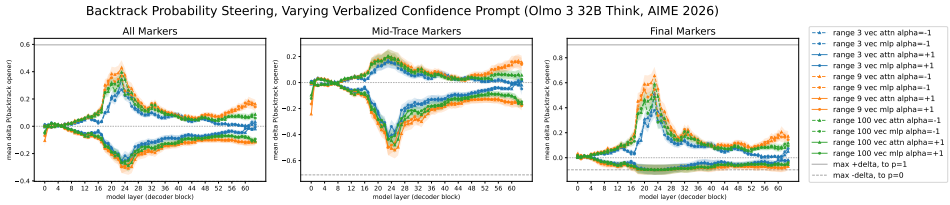
Lastly, we measure how well each confidence direction can steer verbalized confidence with a range of 9 on AIME 2026. Figure 3a shows the steering performance of the DIM vectors across all layers. The Difference in Means direction obtained directly from the range 9 scale saturates the positive headroom and remains close to the negative headroom from the middle layers onward. The other two DIM directions perform equally well in the middle to late layers, with performance tapering near the end, likely due to the lexical differences in the prompts finally taking over. We also get the performance of the range 100 and range 3 vectors on AIME 2026, and find similar performance to the range 9 vector, so we stick with range 9 for the rest of the experiments. See Figure 3b.

	Olmo 3 32B		Qwen 3 32B		Distill 32B	
	Spearman	CKA	Spearman	CKA	Spearman	CKA
Range 9, Range 100	0.978	0.967	0.969	0.961	0.979	0.958
Range 3, Range 100	0.910	0.835	0.962	0.856	0.942	0.962
Range 3, Range 9	0.907	0.827	0.958	0.852	0.939	0.969
Average	0.932	0.876	0.963	0.890	0.953	0.963

Table 14: Spearman correlation and CKA between representations across digit ranges.



(a) Verbalized confidence steering on AIME 2026 across verbalized confidence prompts



(b) p_{bt} steering on Aime 2026 across verbalized confidence prompts

Figure 3: Steering Verbalized Confidence and p_{bt} for Range 3, Range 9, and Range 100 confidence directions

D PROBABILITY DIM ABLATION EXPERIMENT

v_{conf} being able to steer p_{bt} may trivially be explained as the verbalized confidence prompt simply decoding p_{bt} from the last token position of the answer proposal. To separate this effect from v_{conf}, we construct v_{prob} — a DiM direction separating high and low p_{bt} classes — and measure its performance on VC steering. Since a histogram of the proposal markers binned by p_{bt} is bimodal, concentrated near 0-0.1 and 0.9-1, we separate markers into high and low probability classes based on a threshold of 0.5. Each model's v_{prob} is computed from the same dataset that produced the best v_{conf}. As shown in Table 16, the results of this probability direction on steering p_{bt} is the best among the three directions. However, it performs poorly on steering verbalized confidence, though slightly better than the commitment direction. This demonstrates that the latent confidence representation is not just a trivial decoder for answer proposal next token probability.

D.1 POSITION ABLATION

In §3.2, we established a positive correlation between verbalized confidence at a proposal and whether the proposal is a mid-trace or final marker. Intuitively, within a single rollout, the model's verbalized confidence increases as the model grows in confidence as it explores the possible solution space and accumulates evidence. Given this correlation, selecting high and low confidence markers to construct the direction may amount to noisily selecting late-position vs. early-position markers, in which case v_{conf} would be representing position rather than confidence.

To ablate positional information, we construct a confidence direction, $v_{\text{conf-pos}}$, such that the markers selected for the high and low confidence classes share the same positional distribution. We index the markers of rollout r by their order in the text, $j \in \{0, \ldots, n_r - 1\}$, and define the normalized position $\pi = j/(n_r - 1)$, with $\pi \equiv 1$ for single-marker rollouts. We define the *positional drift* of a fitted direction as the difference in the mean normalized position between its two classes, $\bar{\pi}_{\text{high}} - \bar{\pi}_{\text{low}}$, measured over the markers that entered the fit. 0 indicates the classes are drawn from the same place in the trace. We then assign the markers in the train split into four positional bins: first marker, early half, late half, and final marker. If a rollout only contains one marker, it is assigned as a final marker. Thus, each marker is assigned a positional bin and a binary confidence label.

Let H_i and L_i be the number of high and low confidence markers in position bin i. When selecting markers, bin i can supply at most $U_i = \min(H_i, L_i)$ markers of each class since supplying more than U_i would mean one class contains more markers from that bin than the other. From bin i, we draw $k_i = \text{round}(Np_i)$ samples of each class with $p_i = U_i / \sum_j U_j$ and $N = \min(100, \sum_i U_i)$. Intuitively, bins that contain fewer markers should also provide fewer markers to construct the DiM vector. The k_i calculated based off the train split and fixed before sampling. For each model, we construct $v_{\text{conf-pos}}$ from the dataset split of the best performing v_{conf}.

Enforcing a shared positional distribution based on discrete bins eliminates positional drift between bins by construction. However, slight positional bias within the early middle and late middle bins persists since the classes may still differ in position inside a given bin. Empirically this residual is small: $v_{\text{conf-pos}}$ on average drops 83% of the positional drift relative to v_{conf} (Table 15), while retaining 89.2% of v_{conf}'s steering gap on p_{bt} (Table 16). The confidence direction therefore cannot be explained solely as a position detector.

Table 15 shows the positional drift of $v_{\text{conf-pos}}$ and v_{conf}.

Table 15: Position drift of v_{conf} and $v_{\text{conf-pos}}$. 0 means the two classes are drawn from the same relative proposal positions in the trace.

Model	Dataset	v_{conf}	$v_{\text{conf-pos}}$	Reduction
Olmo-3-32B-Think	AIME-2026	$+0.267$	$+0.049$	82%
Qwen3-32B	AIME-2026	$+0.269$	$+0.068$	75%
R1-Distill-Qwen-32B	GSM8K	$+0.264$	$+0.020$	92%
mean		$+0.266$	$+0.046$	83%

E SINGLE TOKEN STEERING

Steering Details We use 50 sampled markers from the validation set. with $\alpha \in \{-1, -0.5, -0.25, 0.25, 0.5, 1\}$. Since the residual stream norm of a model gradually increases across layers, we set α to be in units of the steering site's average residual norm, so the overall intervention is the product of the α and residual norm scalars with the unit direction v_{conf}.

Steering p_{bt} We follow the same procedure as described in §3.2. Since the AIME 2026 has at most 120 final markers (30 prompts $\times$ 4 rollouts, including judge pipeline filtering) split across train, validation, and test, there are fewer than 100 final proposal markers in the train set. Here, we select all available markers for M^+, and keep $|M^-| = 100$.

For the validation sweep, we use the same alpha grid sweep as described in §3.2, and we sample 25 final and 25 mid-trace markers for the validation set.

Table 16: p_{bt} and VC steering with each model's best v_{commit} and v_{conf} on the marker test set. We also test v_{prob} from Appendix D, which is fit on the best dataset for v_{conf}. Gap is the absolute distance between the positive- and negative-steering effects.

		Δp_{bt}					Δ VC				
Model	Vector	Dataset	Site	Negative	Positive	Gap	Dataset	Site	Negative	Positive	Gap
Olmo 3 32B	v_{commit}	GPQA-D	43.attn	+0.258	-0.298	0.556	GSM8K	24.attn	-0.471	+0.609	1.080
	v_{conf}	AIME 2026	23.attn	+0.361	-0.224	0.585	GPQA-D	23.attn	-3.502	+1.215	4.717
	$v_{\text{prob.}}$	AIME 2026	28.attn	+0.486	-0.300	0.786	GPQA-D	23.attn	-0.111	+0.294	0.405
Qwen 3 32B	v_{commit}	GPQA-D	51.mlp	+0.399	-0.242	0.641	GSM8K	48.attn	-0.627	+0.595	1.222
	v_{conf}	AIME 2026	47.attn	+0.423	-0.222	0.645	MMLU-Pro	51.attn	-3.284	+1.002	4.286
	$v_{\text{prob.}}$	AIME 2026	47.attn	+0.681	-0.273	0.954	MMLU-Pro	52.attn	-0.245	+0.481	0.726
Distill 32B	v_{commit}	MMLU-Pro	46.attn	+0.432	-0.382	0.814	AIME 2026	48.mlp	-0.290	+0.273	0.563
	v_{conf}	GSM8K	46.attn	+0.403	-0.341	0.744	AIME 2026	51.attn	-5.198	+0.891	6.089
	$v_{\text{prob.}}$	GSM8K	43.attn	+0.608	-0.337	0.945	AIME 2026	45.attn	-0.556	+0.523	1.079

Best steer site selection criteria Given a model with L layers, we obtain a set of $2L$ directions $V = [v_1, v_2, \ldots, v_{2L}]$ from the DIM formulation, one direction from each post-attention and post-mlp residual across all layers. We ignore the first 15% and last 15% of layers, and select the best site s from the middle 70% of layers. Let E_+, E_- be the maximum effect of positive and negative steering from the candidate layers. Let e^s_+, e^s_- be the effect of positive and negative steering of v_s. We select the best site using the criterion $\text{argmin}_{s,\alpha} C$, where $C = \max_s \left[\min(\frac{e^s_+}{E_+}, \tau) + \min(\frac{e^s_-}{E_-}, \tau)\right]$ with threshold τ. In other words, we select the earliest steering site and smallest α from all s, α combinations that satisfy the criteria C.

When steering p_{bt} with the commitment direction or steering verbalized confidence with the confidence direction, steering performance rises sharply in the middle layers and stays nearly constant towards the final layers. In these cases, to avoid selecting an arbitrarily late layer, we set threshold $\tau = 0.9$. When steering p_{bt} with the confidence direction or steering verbalized confidence with the commitment direction, performance peaks in the middle layers and degrades towards the final layers, so we set a threshold $\tau = 1$.

Full p_{bt} and confidence steering on all models Per dataset p_{bt} and confidence steering validation sweeps are shown in Figure 4 and Figure 5. Steering scores on the marker test set is in Table 17.

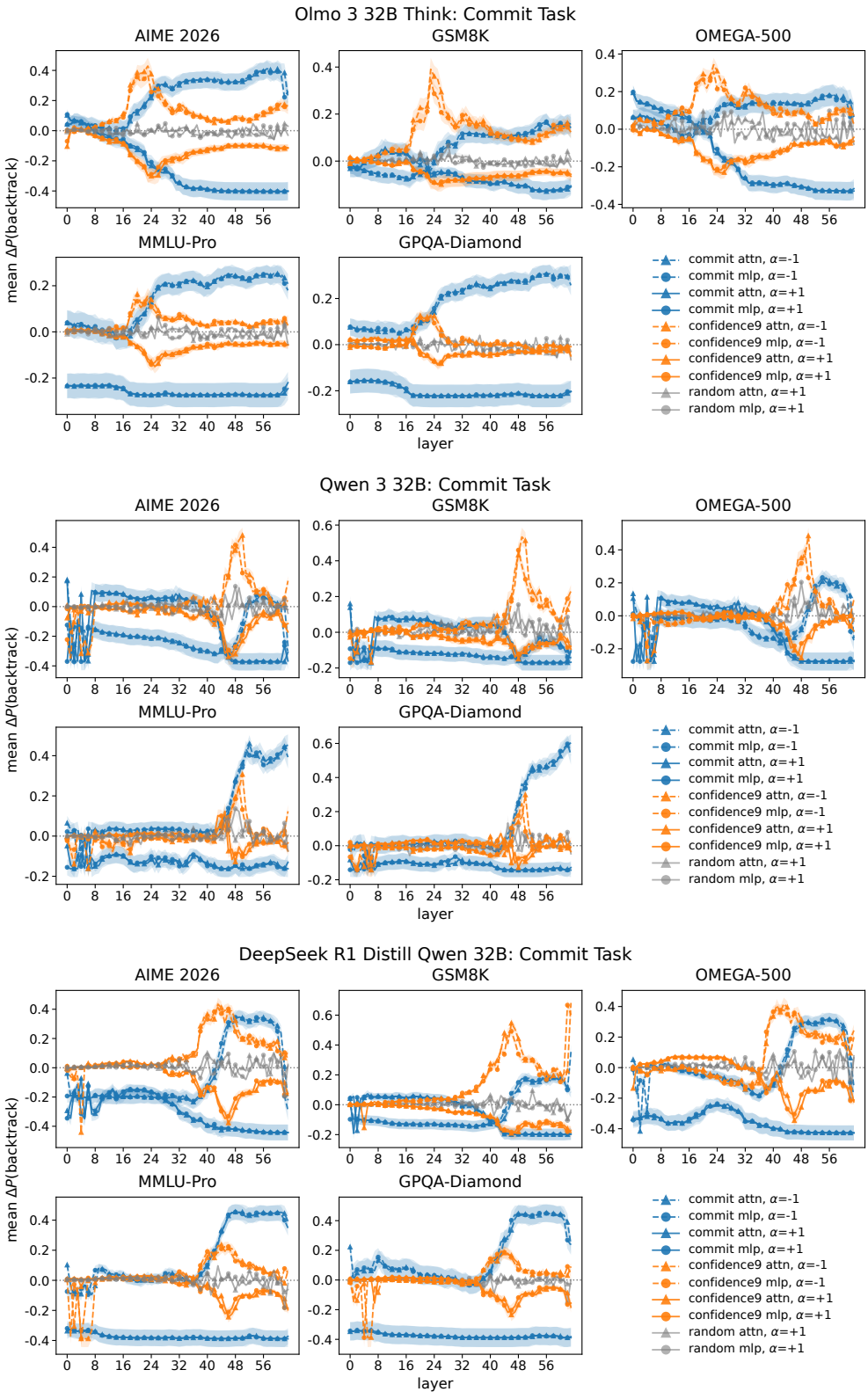


Figure 4: Steering Backtracking Token Results on Olmo 3 32B, Qwen 3 32B, and Distill 32B.

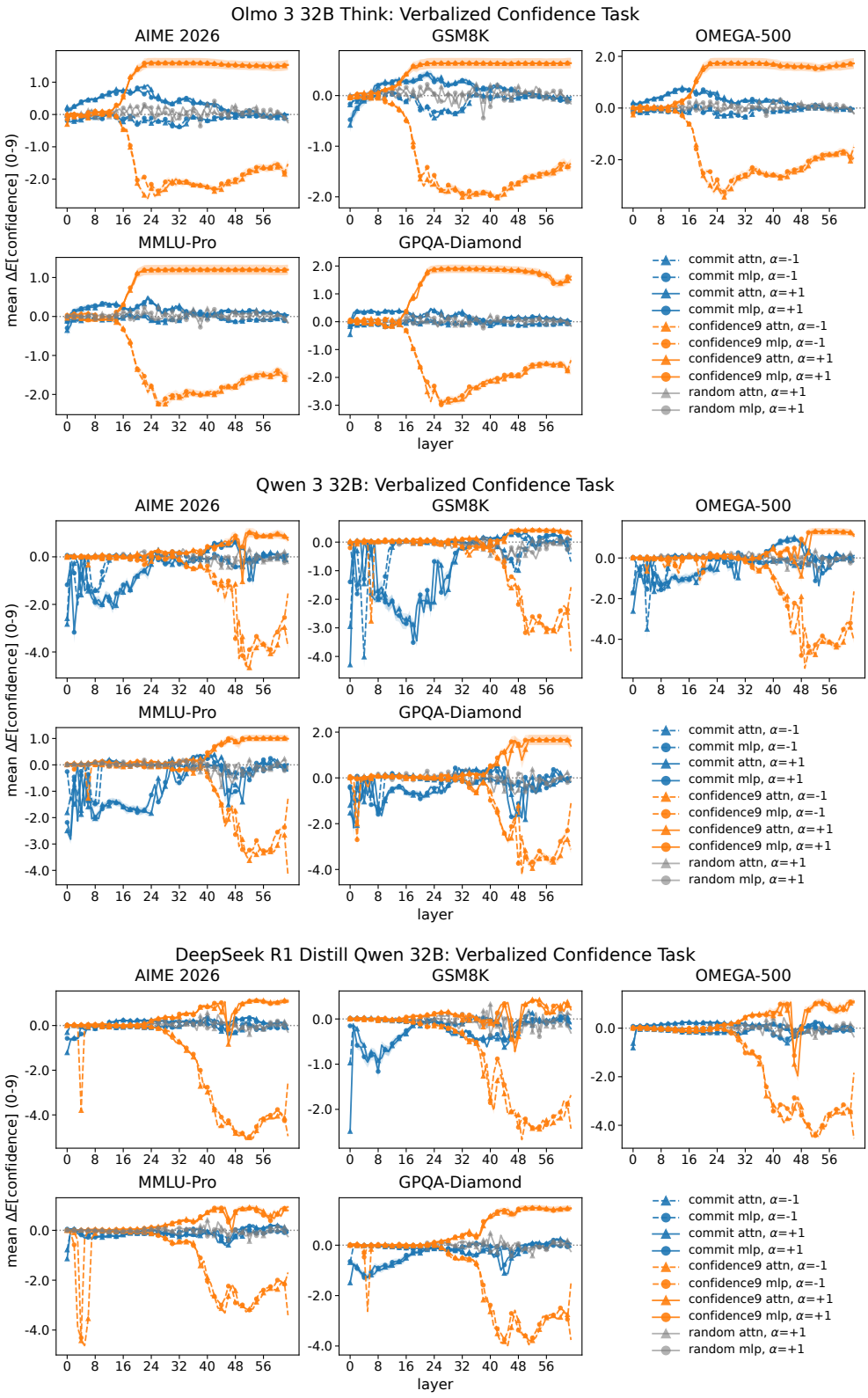


Figure 5: Confidence Steering Results on Olmo 3 32B, Qwen 3 32B, and Distill 32B.

Table 17: Steering on the held-out standard test set. $\Delta P(\text{backtrack})$ is measured at backtrack/commit marker positions; ΔE is the change in next-token entropy over the confidence probe, reported overall and per confidence tercile (the position axis is collapsed by an n-weighted average over the three position roles). Negative/Positive denote the sign of the steering coefficient α; $|\alpha| = 1$, except that * next to a site marks $|\alpha| = 0.5$ and # marks $|\alpha| = 0.25$. Each task selects its own best site. Within each model and vector type, **bold** marks the strongest score in a column and underline the runner-up, "strongest" meaning furthest along the direction that column's steering moves the metric. "–" marks runs that are not finished.

			ΔP(backtrack)								ΔE								
			Site	Negative			Positive				Site	Negative				Positive			
Model	Vector	Dataset		All	Backtrack	Commit	All	Backtrack	Commit			All	High	Medium	Low	All	High	Medium	Low
Olmo 3 32B	Commit	AIME 2026	35.mlp	+0.209	**+0.070**	+0.359	-0.290	-0.487	**-0.078**	20.mlp	-0.216	-0.131	-0.121	-0.472	**+0.623**	+0.277	**+0.803**	**+0.949**
		GSM8K	53.mlp	+0.127	-0.019	+0.284	-0.287	-0.483	**-0.078**	24.attn	**-0.471**	**-0.452**	**-0.427**	**-0.560**	+0.609	+0.286	+0.780	+0.909
		OMEGA-500	39.attn	+0.049	-0.089	+0.197	-0.265	-0.439	**-0.078**	24.attn*	-0.186	-0.162	-0.166	-0.251	+0.269	+0.174	+0.331	+0.343
		GPQA-D	43.attn	**+0.258**	+0.056	**+0.476**	**-0.298**	**-0.502**	**-0.078**	26.mlp	-0.072	-0.173	+0.040	-0.051	+0.172	+0.144	+0.199	+0.181
		MMLU-Pro	43.attn	+0.204	+0.018	+0.405	-0.286	-0.480	**-0.078**	24.attn*	-0.064	-0.078	-0.010	-0.111	+0.211	+0.134	+0.242	+0.296
	Confidence-9	AIME 2026	23.attn	**+0.361**	**+0.302**	**+0.425**	**-0.224**	**-0.365**	**-0.074**	22.attn	-2.310	-2.409	-2.204	-2.288	+1.162	+0.369	+1.231	+2.349
		GSM8K	25.attn	+0.276	+0.254	+0.298	-0.213	-0.344	-0.073	25.attn	-1.538	-1.961	-1.400	-1.035	**+1.230**	**+0.378**	**+1.276**	**+2.539**
		OMEGA-500	23.attn	+0.242	+0.229	+0.255	-0.169	-0.264	-0.066	23.attn	-3.280	-3.786	-3.088	**-2.710**	+1.229	**+0.378**	+1.275	+2.536
		GPQA-D	25.attn	+0.124	+0.136	+0.111	-0.113	-0.175	-0.048	23.attn	**-3.502**	**-4.141**	**-3.356**	-2.663	+1.215	+0.377	+1.268	+2.494
		MMLU-Pro	23.attn	+0.201	+0.195	+0.207	-0.132	-0.203	-0.055	25.mlp	-2.080	-2.598	-1.901	-1.478	+1.226	+0.377	+1.273	+2.526
Qwen 3 32B	Commit	AIME 2026	51.attn	+0.036	-0.124	+0.207	**-0.246**	**-0.413**	**-0.068**	47.mlp	-0.461	-0.366	-0.512	-0.551	**+0.684**	+0.081	**+0.731**	**+1.536**
		GSM8K	52.attn	-0.052	-0.192	+0.097	**-0.246**	-0.412	**-0.068**	48.attn	**-0.627**	**-0.609**	**-0.691**	**-0.588**	+0.595	**+0.096**	+0.727	+1.202
		OMEGA-500	52.mlp	+0.103	-0.053	+0.269	-0.245	-0.412	**-0.068**	46.attn*	-0.185	-0.082	-0.156	-0.369	+0.413	+0.073	+0.544	+0.785
		GPQA-D	51.mlp	**+0.399**	**+0.186**	**+0.625**	-0.242	-0.406	**-0.068**	44.mlp*	-0.100	-0.050	-0.042	-0.237	+0.223	+0.053	+0.294	+0.404
		MMLU-Pro	51.mlp	+0.359	+0.152	+0.579	-0.236	-0.394	**-0.068**	53.mlp*	-0.133	-0.097	-0.120	-0.200	+0.228	+0.086	+0.441	+0.217
	Confidence-9	AIME 2026	47.attn	**+0.423**	+0.292	**+0.563**	**-0.222**	-0.374	**-0.060**	51.attn*	-2.632	-2.705	-2.844	-2.301	+0.961	+0.090	+0.828	+2.404
		GSM8K	48.attn	+0.385	+0.276	+0.502	**-0.222**	**-0.375**	-0.059	50.mlp	-2.893	-3.040	-3.117	-2.439	+0.960	**+0.111**	+0.860	+2.336
		OMEGA-500	48.attn	+0.354	+0.265	+0.450	-0.211	-0.355	-0.057	49.attn*	-3.178	-2.490	**-3.780**	**-3.576**	+0.902	+0.045	+0.746	+2.348
		GPQA-D	50.attn	+0.347	**+0.295**	+0.402	-0.108	-0.178	-0.034	49.attn*	-3.104	-2.606	-3.630	-3.296	+0.887	+0.057	+0.714	+2.312
		MMLU-Pro	50.attn	+0.306	+0.260	+0.355	-0.123	-0.194	-0.047	51.attn	**-3.284**	**-3.406**	-3.559	-2.814	**+1.002**	+0.108	**+0.862**	**+2.485**
R1 Qwen 32B	Commit	AIME 2026	47.mlp	+0.356	+0.086	+0.646	-0.374	-0.581	-0.152	48.mlp	**-0.290**	**-0.165**	**-0.276**	**-0.476**	**+0.273**	+0.157	**+0.288**	**+0.415**
		GSM8K	48.mlp	-0.053	-0.247	+0.155	-0.377	-0.585	-0.152	44.attn#	-0.071	-0.060	-0.052	-0.108	+0.094	+0.079	+0.091	+0.117
		OMEGA-500	47.mlp	+0.333	+0.068	+0.618	-0.370	-0.572	-0.152	44.attn*	-0.079	-0.066	-0.045	-0.135	+0.180	+0.141	+0.211	+0.197
		GPQA-D	46.mlp	+0.292	+0.018	+0.586	-0.381	-0.593	-0.152	44.attn*	-0.052	-0.073	-0.036	-0.041	+0.201	**+0.174**	+0.252	+0.180
		MMLU-Pro	46.attn	**+0.432**	**+0.157**	**+0.728**	**-0.382**	**-0.595**	**-0.153**	44.attn*	-0.089	-0.102	-0.075	-0.086	+0.193	+0.147	+0.247	+0.194
	Confidence-9	AIME 2026	46.attn	+0.344	+0.262	+0.432	-0.330	-0.505	-0.142	51.attn	**-5.198**	**-5.678**	**-5.268**	**-4.464**	+0.891	+0.288	+0.760	+1.861
		GSM8K	46.attn	**+0.403**	**+0.326**	**+0.485**	**-0.341**	**-0.518**	**-0.150**	52.attn	-2.331	-2.346	-2.344	-2.294	**+1.012**	**+0.413**	**+0.868**	**+1.990**
		OMEGA-500	46.attn	+0.331	+0.264	+0.404	-0.300	-0.455	-0.132	52.attn	-4.403	-4.694	-4.463	-3.937	+0.889	+0.299	+0.762	+1.839
		GPQA-D	45.mlp	+0.332	+0.270	+0.398	-0.266	-0.389	-0.133	43.attn	-3.888	-3.774	-4.088	-3.817	+0.924	+0.333	+0.790	+1.880
		MMLU-Pro	46.attn	+0.295	+0.257	+0.336	-0.288	-0.442	-0.123	43.attn	-2.903	-2.767	-3.034	-2.940	+0.896	+0.303	+0.764	+1.854

Table 18: Steering on the held-out standard test set, broken out by source dataset. The test set pools 100 markers from each of the five datasets; each column is the pooled score ($\Delta P(\text{backtrack})$ over all markers, ΔE over all confidence positions) restricted to one of them. Rows are indexed by the dataset the steering vector was fit on. Negative/Positive denote the sign of the steering coefficient α; $|\alpha| = 1$, except that * next to a site marks $|\alpha| = 0.5$ and # marks $|\alpha| = 0.25$. Each task selects its own best site. Within each model and vector type, **bold** marks the strongest score in a column and underline the runner-up, "strongest" meaning furthest along the direction that column's steering moves the metric. "–" marks runs that are not finished.

			ΔP(backtrack)										ΔE									
			Site	Negative					Positive				Site	Negative					Positive			
Model	Vector	Train set		AIME 2026	GSM8K	OMEGA-500	GPQA-D	MMLU-Pro	AIME 2026	GSM8K	OMEGA-500	GPQA-D	MMLU-Pro		AIME 2026	GSM8K	OMEGA-500	GPQA-D	MMLU-Pro	AIME 2026	GSM8K	OMEGA-500	GPQA-D	MMLU-Pro
Olmo 3 32B	Commit	AIME 2026	35.mlp	**+0.225**	+0.161	+0.316	+0.182	+0.162	-0.437	**-0.128**	-0.405	-0.242	-0.236	20.mlp	-0.184	-0.093	-0.246	-0.352	-0.294	**+0.784**	**+0.319**	**+0.711**	+0.734	**+0.567**
		GSM8K	53.mlp	+0.116	+0.117	+0.188	+0.115	+0.099	-0.429	-0.120	-0.403	-0.243	-0.243	24.attn	**-0.431**	**-0.414**	**-0.461**	**-0.556**	**-0.495**	+0.754	+0.241	+0.709	**+0.788**	+0.551
		OMEGA-500	39.attn	+0.044	+0.036	+0.128	+0.028	+0.008	-0.386	-0.110	-0.379	-0.228	-0.221	24.attn*	-0.178	-0.122	-0.169	-0.270	-0.193	+0.327	+0.155	+0.267	+0.339	+0.257
		GPQA-D	43.attn	+0.108	**+0.187**	**+0.326**	**+0.297**	**+0.283**	**-0.456**	-0.125	**-0.409**	**-0.254**	**-0.245**	26.mlp	-0.026	-0.081	-0.079	-0.087	-0.084	+0.244	+0.067	+0.153	+0.254	+0.139
		MMLU-Pro	43.attn	+0.162	+0.134	+0.283	+0.227	+0.214	-0.432	-0.106	-0.401	-0.246	**-0.245**	24.attn*	-0.030	-0.051	-0.052	-0.125	-0.062	+0.283	+0.109	+0.235	+0.261	+0.168
	Confidence-9	AIME 2026	23.attn	**+0.423**	**+0.406**	**+0.464**	**+0.259**	**+0.252**	**-0.332**	-0.103	**-0.325**	**-0.178**	-0.184	22.attn	-2.621	-2.141	-2.726	-2.005	-2.058	+1.143	+0.471	+1.445	+1.675	+1.078
		GSM8K	25.attn	+0.314	+0.300	+0.351	+0.194	+0.219	-0.301	**-0.104**	-0.289	**-0.178**	**-0.196**	25.attn	-1.576	-1.900	-1.527	-1.209	-1.478	**+1.175**	**+0.491**	+1.497	**+1.848**	**+1.138**
		OMEGA-500	23.attn	+0.288	+0.272	+0.301	+0.157	+0.190	-0.256	-0.069	-0.226	-0.142	-0.151	23.attn	-3.383	-3.763	-3.382	-2.768	-3.103	+1.174	+0.490	**+1.498**	+1.844	+1.137
		GPQA-D	25.attn	+0.141	+0.189	+0.107	+0.085	+0.099	-0.166	-0.058	-0.128	-0.098	-0.116	23.attn	**-3.542**	**-4.089**	**-3.532**	**-2.916**	**-3.433**	+1.167	+0.486	+1.488	+1.809	+1.127
		MMLU-Pro	23.attn	+0.233	+0.233	+0.256	+0.127	+0.156	-0.196	-0.054	-0.164	-0.117	-0.128	25.mlp	-2.045	-2.615	-2.021	-1.677	-2.044	+1.170	+0.489	+1.487	+1.844	+1.137
Qwen 3 32B	Commit	AIME 2026	51.attn	+0.052	-0.036	+0.106	+0.003	+0.055	**-0.372**	**-0.161**	**-0.303**	**-0.251**	**-0.142**	47.mlp	-0.379	-0.505	-0.417	-0.471	-0.534	+0.680	+0.298	+0.888	**+0.921**	**+0.633**
		GSM8K	52.attn	-0.030	-0.089	+0.002	-0.102	-0.043	-0.371	**-0.161**	**-0.303**	**-0.251**	**-0.142**	48.attn	**-0.527**	**-0.859**	**-0.541**	**-0.542**	**-0.667**	**+0.758**	**+0.345**	**+0.904**	+0.740	+0.226
		OMEGA-500	52.mlp	+0.140	+0.023	+0.163	+0.067	+0.120	-0.371	**-0.161**	**-0.303**	-0.250	-0.141	46.attn*	-0.176	-0.139	-0.188	-0.215	-0.207	+0.440	+0.265	+0.523	+0.475	+0.362
		GPQA-D	51.mlp	**+0.378**	**+0.302**	**+0.409**	**+0.422**	**+0.483**	-0.371	-0.156	**-0.303**	-0.246	-0.137	44.mlp*	-0.117	-0.079	-0.084	-0.116	-0.105	+0.286	+0.171	+0.291	+0.218	+0.151
		MMLU-Pro	51.mlp	+0.357	+0.247	+0.385	+0.380	+0.425	-0.365	-0.141	-0.301	-0.234	-0.141	53.mlp*	-0.144	-0.036	-0.148	-0.192	-0.143	+0.288	+0.214	+0.281	+0.146	+0.210
	Confidence-9	AIME 2026	47.attn	**+0.491**	**+0.463**	**+0.449**	**+0.302**	**+0.411**	**-0.349**	**-0.144**	-0.270	-0.229	-0.119	51.attn*	-2.621	-2.608	-2.552	-2.599	-2.780	+0.895	+0.420	+1.366	+1.288	+0.836
		GSM8K	48.attn	+0.434	+0.454	+0.396	+0.286	+0.355	-0.340	-0.137	-0.270	**-0.231**	**-0.132**	50.mlp	-2.940	-2.975	-2.783	-2.783	-2.986	+0.811	**+0.483**	+1.322	+1.312	**+0.873**
		OMEGA-500	48.attn	+0.417	+0.412	+0.350	+0.251	+0.342	-0.333	-0.130	**-0.281**	-0.186	-0.124	49.attn*	**-3.415**	-2.689	**-3.306**	-3.230	-3.251	+0.781	+0.333	+1.295	+1.281	+0.818
		GPQA-D	50.attn	+0.335	+0.449	+0.324	+0.272	+0.354	-0.115	-0.061	-0.125	-0.138	-0.102	49.attn*	-3.342	-2.700	-3.225	-3.071	-3.179	+0.848	+0.195	+1.270	+1.304	+0.821
		MMLU-Pro	50.attn	+0.300	+0.391	+0.285	+0.248	+0.305	-0.153	-0.087	-0.141	-0.134	-0.101	51.attn	-3.205	**-3.481**	-2.979	**-3.246**	**-3.511**	**+0.928**	+0.481	**+1.413**	**+1.313**	**+0.873**
R1 Qwen 32B	Commit	AIME 2026	47.mlp	+0.295	+0.358	+0.341	+0.429	+0.356	-0.450	-0.244	-0.386	-0.430	-0.363	48.mlp	**-0.292**	**-0.171**	**-0.330**	**-0.377**	**-0.279**	**+0.342**	+0.117	**+0.260**	**+0.359**	**+0.289**
		GSM8K	48.mlp	-0.135	+0.015	-0.074	-0.026	-0.045	-0.456	-0.245	-0.386	-0.429	-0.362	44.attn#	-0.087	-0.055	-0.086	-0.072	-0.056	+0.135	+0.057	+0.112	+0.090	+0.075
		OMEGA-500	47.mlp	+0.259	+0.357	+0.313	+0.403	+0.333	-0.446	-0.237	-0.383	-0.428	-0.355	44.attn*	-0.052	-0.081	-0.087	-0.102	-0.071	+0.239	+0.078	+0.189	+0.218	+0.175
		GPQA-D	46.mlp	+0.215	+0.317	+0.275	+0.353	+0.300	**-0.460**	-0.247	-0.390	**-0.434**	**-0.372**	44.attn*	-0.002	-0.078	-0.052	-0.067	-0.060	+0.249	**+0.118**	+0.217	+0.233	+0.188
		MMLU-Pro	46.attn	**+0.357**	**+0.491**	**+0.409**	**+0.468**	**+0.435**	**-0.460**	**-0.248**	**-0.394**	**-0.434**	**-0.372**	44.attn*	-0.027	-0.110	-0.085	-0.117	-0.106	+0.230	+0.096	+0.196	+0.248	+0.194
	Confidence-9	AIME 2026	46.attn	+0.382	+0.390	+0.412	+0.280	+0.256	-0.397	-0.212	**-0.342**	-0.377	-0.324	51.attn	**-4.836**	**-5.704**	**-5.213**	**-4.922**	**-5.315**	+1.270	+0.313	+0.911	+1.204	+0.757
		GSM8K	46.attn	**+0.418**	**+0.525**	**+0.461**	**+0.310**	**+0.300**	**-0.416**	**-0.218**	-0.339	**-0.391**	**-0.341**	52.attn	-2.239	-2.440	-2.395	-2.254	-2.325	**+1.410**	**+0.460**	**+1.071**	**+1.279**	**+0.838**
		OMEGA-500	46.attn	+0.353	+0.379	+0.399	+0.274	+0.252	-0.360	-0.192	-0.310	-0.336	-0.299	52.attn	-4.141	-4.717	-4.466	-4.222	-4.467	+1.274	+0.315	+0.919	+1.193	+0.747
		GPQA-D	45.mlp	+0.366	+0.374	+0.393	+0.275	+0.251	-0.318	-0.176	-0.276	-0.292	-0.267	43.attn	-3.784	-3.711	-3.999	-3.897	-4.048	+1.320	+0.369	+0.988	+1.185	+0.755
		MMLU-Pro	46.attn	+0.335	+0.317	+0.359	+0.244	+0.221	-0.352	-0.178	-0.302	-0.323	-0.286	43.attn	-2.803	-2.773	-3.016	-2.918	-3.007	+1.286	+0.333	+0.952	+1.172	+0.736

F CONFIDENCE STEERING ON LINGUISTIC CONFIDENCE

Table 19: Effect of regenerating an answer proposal with $\alpha = +0.5$ confidence steering on Olmo 3 32B (AIME 2026, problem index 27). Both continuations share the same prefix. Adding confidence causes the model to consolidate and commit.

Problem. Call finite sets of integers S and T *cousins* if (i) S and T have the same number of elements, (ii) S and T are disjoint, and (iii) the elements of S can be paired with the elements of T so that the elements in each pair differ by exactly 1. For example, $\{1, 2, 5\}$ and $\{0, 3, 4\}$ are cousins. Suppose that the set S has exactly 4040 cousins. Find the least number of elements the set S can have. **Ground Truth.** 107.	
Shared prefix (through marker 29). *Thus $S = \{0, 2, 3\}$ has only 1 cousin. Not helpful. This isn't better.*	
Original (Unsteered)	**Regenerated (Steered $\alpha = 0.5$)**
Given that I can't find a better configuration, I think the answer is 107. But since this is an Olympiad problem, maybe the answer is smaller. Perhaps my initial assumption about...	*After exploring various configurations and consistently arriving at 107 as the minimal, I think that's the answer. So I'll go with* $\boxed{107}$ ***Final Answer*** $\boxed{107}$ `</think>` *To solve the problem...*
Proposes 107, then immediately backtracks.	Proposes 107, affirms it, and exits the reasoning trace.

Table 20: Effect of regenerating an answer proposal with $\alpha = -0.5$ confidence steering on Qwen3-32B (GPQA-D, problem index 47). Both continuations share the same prefix. Removing confidence causes the model to revisit other options.

Problem. Cyclohexanone was treated with bromine, forming product 1. 1 was heated with sodium hydroxide, forming product 2. 2 was treated with thionyl chloride and pyridine, forming 3. 3 was treated with lithium tri-*tert*-butoxyaluminum hydride, forming product 4. How many chemically distinct hydrogen atoms are there on product 4? **Ground Truth.** D. (A) 10 (B) 7 (C) 8 (D) 6	
Shared prefix (through marker 3). *But let me verify once more. If product 4 is cyclopentanol, with 6 distinct hydrogen environments, then answer D (6) is correct.*	
Unsteered	**Steered**
Therefore, the answer is D. `</think>` *To determine the...*	*But let's think about another possibility: maybe product 4 is not cyclopentanol. Let me re-examine the steps.* *Product 2: cyclopentanecarboxylic acid? Or something else?...*
Commits to D and exits the reasoning block.	Reopens the question and revisits an earlier intermediate.

Table 21: Effect of regenerating an answer proposal with $\alpha = -0.5$ confidence steering on Distill 32B (GSM8K, problem index 785). Both continuations share the same prefix. Negative steering causes the model to re-checks the problem statement and reopen computation.

Problem. Carrie was given ten twenties and 140 quarters by her aunt to use for lunch. If she spent all the quarters and $3/5$ of the twenties, calculate the total amount of money she paid for the lunch. **Ground Truth.** 155.

Shared prefix (through marker 4). *Adding those together, 35 + 120 is 155. Yep, that seems accurate.*

Unsteered	**Steered**
So, I think I'm confident that Carrie spent a total of $155 on lunch. ***Final Answer*** *The total amount Carrie paid for lunch is* $\boxed{155}$ *dollars.* `</think>` *Carrie was given ten twenties and 140 quarters by her aunt. To find the total amount...*	*So, I think the total amount Carrie paid for lunch is $155. Let me just make sure I didn't miss anything in the problem. She was given ten twenties and 140 quarters. She spent all the quarters and $3/5$ of the twenties. So, I think I accounted for that correctly.* *Wait, hold on, let me think again...*
States the answer and exits the reasoning block.	Re-reads the problem, then reopens the computation.

G Multi Token Steering Miscellaneous

G.1 CoT vs Steering metrics

In Table 23, we record the baseline `</think>` / p_{bt} used in Table 3.

Table 22: Confident and Uncertain Sentences used in the surface-level vs latent confidence experiments. Let `X` be the answer value. We do not replace the final punctuation of the sentence, but rather use the same punctuation token that the original answer proposal sentence uses.

Sentence Type	Sentence
Confident Sentence 1	`Therefore, I am confident the answer is X`
Confident Sentence 2	`Therefore, my final answer is X`
Confident Sentence 3	`Therefore, the answer is \boxed {X}`
Uncertain Sentence 1	`I am not confident, but I think the answer is X`
Uncertain Sentence 2	`Therefore, the answer may be X`
Uncertain Sentence 3	`Perhaps the answer is X`

Table 23: `</think>` / p_{bt} baselines for surface-level vs latent confidence results in Table 3. The Δs in Table 3 are computed off each method's own baseline values The slight numerical variations are due to bfloat16 vLLM implementation differences. n is the total number of rollouts generated (four per marker) from the markers that are survive filtering.

Model	Metric	Positive: confident proposal (mid-trace markers)				Negative: hedged proposal (commitment markers)			
		transplant	append	steer (α=0)	n	transplant	append	steer (α=0)	n
OLMo-3-32B	`</think>`	3.2	3.5	3.5	564	97.2	97.1	97.3	792
	p_{bt}	84.8	84.8	85.9		4.9	4.9	5.1	
Qwen3-32B	`</think>`	6.8	6.8	6.8	456	97.7	97.7	97.7	684
	p_{bt}	82.9	82.9	83.0		5.4	5.4	5.9	
R1-Qwen-32B	`</think>`	11.2	11.4	11.1	668	91.7	91.7	91.7	544
	p_{bt}	82.8	82.8	82.9		13.9	13.9	13.9	

H Post Marker Steering Details

Table 24: Post-marker continuation steering on the pooled 500-marker `standard_test_set.json`. ΔL is generated length minus the stored unsteered rollout's remaining length; C→I and I→C count answers that flipped against that same rollout, and Δacc is their net, in points. Δprop is the mean change in the number of judged answer proposals after the marker, steered continuation minus the stored rollout; – where the run has not been judged. ΔL and Δprop are given for all markers (mean), mid-trace (backtrack) markers and the final marker of each rollout (mean$_{\pm\mathrm{SE}}$, standard error of the mean clustered by source rollout). short is the fraction of markers whose steered continuation is shorter than the stored rollout's remainder.

Model	Vec	α	$\overline{\Delta L}$			Δprop			short	C→I	I→C	Δacc
			all	mid-trace	final	all	mid-trace	final				
OLMo-3-32B	commit	-0.5	901	$1175_{\pm 156}$	$607_{\pm 115}$	$+6.10$	$+8.86_{\pm 1.34}$	$+3.16_{\pm 0.51}$	0.30	11	7	-0.8
		-0.25	336	$381_{\pm 100}$	$286_{\pm 100}$	–	–	–	0.39	3	1	-0.4
		-0.1	98	$75_{\pm 94}$	$122_{\pm 37}$	–	–	–	0.45	10	5	-1.0
		0.1	-155	$-310_{\pm 102}$	$12_{\pm 10}$	–	–	–	0.52	6	5	-0.2
		0.25	-373	$-782_{\pm 148}$	$67_{\pm 51}$	-2.53	$-5.07_{\pm 0.66}$	$+0.17_{\pm 0.04}$	0.62	11	5	-1.2
		0.5	3476	$3674_{\pm 741}$	$3264_{\pm 499}$	–	–	–	0.56	92	2	-18.0
		0.75	20517	$20611_{\pm 615}$	$20417_{\pm 624}$	–	–	–	0.05	351	0	-70.2
	conf	-0.5	8144	$11602_{\pm 428}$	$4426_{\pm 376}$	–	–	–	0.06	41	5	-7.2
		-0.25	1959	$3180_{\pm 186}$	$647_{\pm 111}$	$+3.18$	$+4.78_{\pm 1.00}$	$+1.49_{\pm 0.29}$	0.15	13	6	-1.4
		-0.1	543	$842_{\pm 110}$	$222_{\pm 60}$	–	–	–	0.33	6	6	+0.0
		0.1	-423	$-808_{\pm 102}$	$-9_{\pm 9}$	–	–	–	0.65	8	4	-0.8
		0.25	-728	$-1379_{\pm 122}$	$-30_{\pm 9}$	–	–	–	0.76	7	3	-0.8
		0.5	-978	$-1827_{\pm 143}$	$-64_{\pm 11}$	–	–	–	0.84	11	5	-1.2
		0.75	-1099	$-2030_{\pm 147}$	$-98_{\pm 13}$	-3.67	$-7.19_{\pm 0.68}$	$+0.13_{\pm 0.03}$	0.89	13	5	-1.6
Qwen3-32B	commit	-0.5	23015	$23012_{\pm 455}$	$23019_{\pm 483}$	–	–	–	0.00	71	4	-17.9
		-0.25	521	$708_{\pm 85}$	$321_{\pm 36}$	$+3.75$	$+5.00_{\pm 0.60}$	$+2.42_{\pm 0.23}$	0.24	7	7	+0.0
		-0.1	162	$207_{\pm 99}$	$115_{\pm 22}$	–	–	–	0.38	4	8	+0.8
		0.1	-98	$-213_{\pm 98}$	$24_{\pm 19}$	–	–	–	0.55	2	7	+1.0
		0.25	-249	$-487_{\pm 108}$	$5_{\pm 11}$	-1.66	$-3.24_{\pm 0.39}$	$+0.03_{\pm 0.01}$	0.61	4	9	+1.0
		0.5	-699	$-1129_{\pm 232}$	$-241_{\pm 27}$	–	–	–	0.82	98	4	-18.8
		0.75	1968	$1749_{\pm 615}$	$2202_{\pm 526}$	–	–	–	0.83	317	2	-63.0
	conf	-0.5	14395	$19604_{\pm 714}$	$8841_{\pm 767}$	–	–	–	0.13	205	5	-40.0
		-0.25	1304	$2102_{\pm 262}$	$453_{\pm 68}$	$+2.87$	$+4.05_{\pm 0.51}$	$+1.63_{\pm 0.22}$	0.24	14	11	-0.6
		-0.1	325	$531_{\pm 124}$	$105_{\pm 29}$	–	–	–	0.36	6	9	+0.6
		0.1	-187	$-400_{\pm 85}$	$39_{\pm 14}$	–	–	–	0.55	4	7	+0.6
		0.25	-321	$-667_{\pm 115}$	$48_{\pm 14}$	-1.25	$-2.60_{\pm 0.44}$	$+0.19_{\pm 0.08}$	0.59	6	4	-0.4
		0.5	-208	$-493_{\pm 287}$	$97_{\pm 67}$	–	–	–	0.63	21	11	-2.0
		0.75	1395	$2150_{\pm 557}$	$590_{\pm 243}$	–	–	–	0.63	74	12	-12.4
R1-Qwen-32B	commit	-0.25	9712	$13647_{\pm 836}$	$5483_{\pm 630}$	–	–	–	0.10	95	3	-18.4
		-0.1	2600	$3917_{\pm 635}$	$1184_{\pm 288}$	$+3.10$	$+4.21_{\pm 0.63}$	$+2.07_{\pm 0.29}$	0.28	22	4	-3.6
		0.1	-144	$-310_{\pm 121}$	$35_{\pm 11}$	–	–	–	0.54	8	7	-0.2
		0.25	-353	$-688_{\pm 89}$	$7_{\pm 8}$	–	–	–	0.66	6	5	-0.2
		0.5	-553	$-1076_{\pm 93}$	$9_{\pm 11}$	-2.98	$-5.79_{\pm 0.48}$	$+0.01_{\pm 0.01}$	0.75	10	6	-0.8
		0.75	2646	$2278_{\pm 597}$	$3042_{\pm 573}$	–	–	–	0.66	66	6	-12.0
	conf	-0.25	6294	$8602_{\pm 693}$	$3813_{\pm 483}$	–	–	–	0.11	53	11	-8.4
		-0.1	1587	$2436_{\pm 526}$	$675_{\pm 164}$	$+2.10$	$+2.40_{\pm 0.59}$	$+1.80_{\pm 0.42}$	0.31	16	6	-2.0
		0.1	22	$-82_{\pm 212}$	$135_{\pm 96}$	–	–	–	0.57	10	9	-0.2
		0.25	-350	$-654_{\pm 141}$	$-23_{\pm 8}$	–	–	–	0.72	9	7	-0.4
		0.5	-530	$-985_{\pm 91}$	$-41_{\pm 11}$	-1.90	$-3.78_{\pm 0.48}$	$+0.15_{\pm 0.03}$	0.78	9	6	-0.6
		0.75	277	$168_{\pm 421}$	$395_{\pm 232}$	–	–	–	0.78	23	7	-3.2

I FULL LOGIT LENS

Table 25: Unfiltered top-10 logit lens tokens for the negative and positive steering directions, sorted in descending order by logit value. Leading spaces are part of the token. Across all three models the negative direction is dominated by hedging and minimizing expressions, in English and in Chinese, while the positive direction favors certainty and completeness markers.

Model	Direction	Top-10 logit lens tokens
Olmo 3 32B	Negative	' somewhat', ' questionable', ' controversial', ' slightly', ' vaguely', ' uncertain', ' unclear', '-ish', ' unfortunately', '=False'
	Positive	' strongest', 'ses', 'zes', ' perfected', ' superior', '[U+FFFD]', ' stronger', ' flawless', ' perfectly', '/all'
Qwen 3 32B	Negative	'有一定的', '赦', '具有一定', '部分内容', '悬', '鹊', '幕后', '有一定', '主要领导', '好吗'
	Positive	'ourt', '无疑是', '完全可以', 'ock', '毫无疑', '完美的', ' следует', '享有', '赞', '做强'
R1-Qwen-32B	Negative	'不一定', '不确定', '不是很', '勉强', '未必', ' uncertain', '一半', '初步', '不如', '有一定的'
	Positive	'放心', ' flawless', ' redund', 'Trivia', '´ăng', '康', 'ubern', '无疑', '久', '亲自'