RETHINKING STABILITY FOR REASONING PATHS IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) demonstrate remarkable capabilities in complex reasoning tasks, where test-time compute has emerged as a crucial means for performance enhancement. Existing methods mainly favor outputs supported by answer consistency or high token confidence, implicitly treating stable reasoning paths as more reliable. However, these approaches provide limited guidance on when stability is a reliable selection signal. In this study, we analyze how the relationship between token stability and correctness varies with the model's confidence in its reasoning path. Our analysis shows that stability is informative when the model is confident, whereas unstable paths should not be uniformly discarded when the model is uncertain. Motivated by this finding, we introduce the **Verbalized-Token Synergistic Filter** (**VeTo**), which combines macro-level verbalized confidence with micro-level token confidence fluctuations to filter reasoning paths across confidence regimes. Comprehensive evaluations across eight diverse benchmarks spanning mathematical, multidisciplinary, code, and multimodal reasoning tasks on three model families (Qwen, Llama, and InternLM) demonstrate that VeTo achieves the highest average accuracy within each model family among all compared strategies. Crucially, our method scales robustly with inference budgets from $N = 4$ to 64 and across model sizes from 4B to 14B, while also yielding competitive calibration performance in mathematical reasoning. Overall, our findings provide a new perspective on reasoning path selection by interpreting stability and instability according to model confidence.

1 INTRODUCTION

Advanced reasoning is a critical capability for large language models (LLMs), enabling them to address complex problems in areas such as scientific discovery and disease diagnosis (Lu et al., 2026; Zhao et al., 2026). Recent test-time scaling methods further improve this capability by allocating additional computation during inference (Snell et al., 2024; Wu et al., 2024). Test-time compute can be allocated either to deepen a single reasoning process or to broaden the search space through parallel sampling (Muennighoff et al., 2025; Brown et al., 2024). Within the latter paradigm, existing methods aggregate or select among generated outputs using different signals. For instance, consensus-driven methods such as Self-Consistency aggregate multiple outputs to mitigate generation variance, while token-based selection techniques prioritize high-confidence trajectories (Wang et al., 2022; 2024; Fu et al., 2026; Kang et al., 2025). Despite their different selection criteria, these methods favor outputs supported by answer consistency or high token confidence (Wang et al., 2022; Fu et al., 2026). However, these signals alone provide limited guidance on when token-level stability is a reliable selection criterion and when unstable paths remain worth retaining. The central challenge is therefore to benefit from stability without retaining stable errors or discarding valid but unstable solutions.

To illustrate this challenge, we conduct controlled experiments on token confidence dynamics using GSM8K and MATH500. We construct the datasets from reasoning paths that yield correct answers to reduce confounding from problems the models cannot solve. Specifically, we retain the initial portion of a correct reasoning path, append a deliberately erroneous step, and truncate the remaining solution to form a misleading prefix. We then ask the models to continue reasoning from the resulting prefixes. Construction details and representative examples are provided in Appendix Section A.1. As shown in Figure 1, paths that produce incorrect answers cluster in the high average

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.

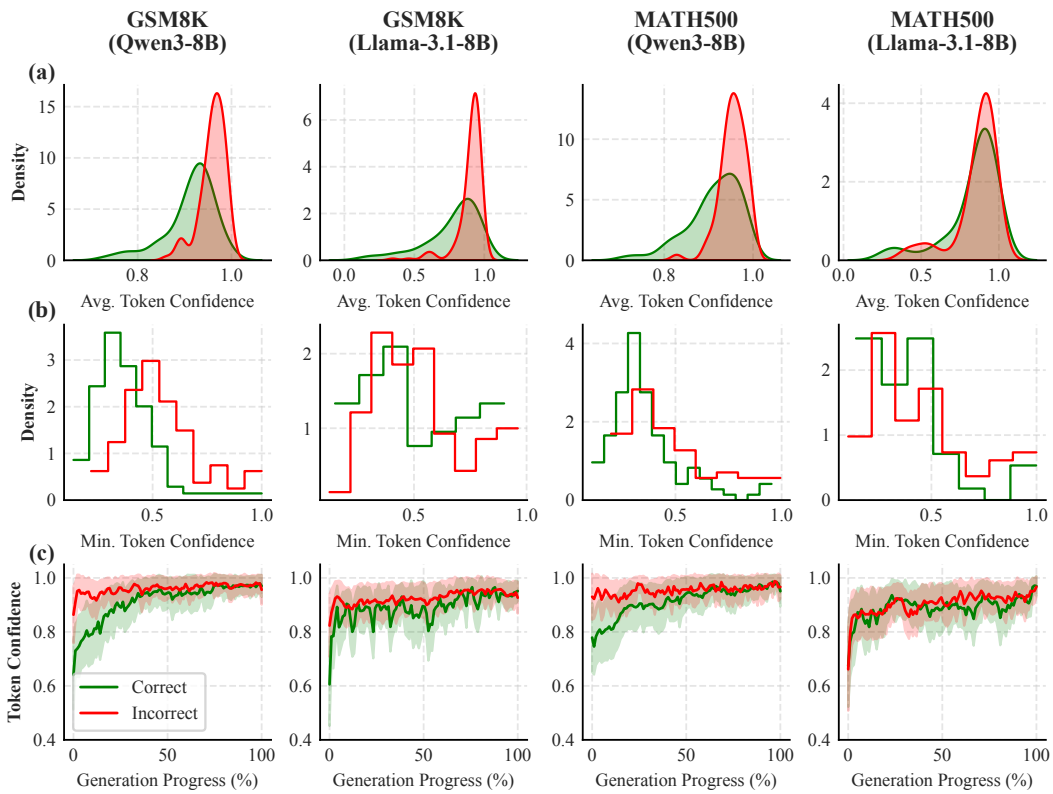


Figure 1: **Token confidence distributions and dynamics under misleading prefixes.** Results are shown for Qwen3-8B and Llama-3.1-8B on GSM8K and MATH500. **(a)** Average token confidence distributions. **(b)** Minimum token confidence distributions. **(c)** Confidence trajectories over generation progress. Green and red denote correct and incorrect continuations, respectively.

token confidence regime, whereas those that produce correct answers are more dispersed and more frequently reach low minimum token confidence. Additionally, correct paths can exhibit pronounced fluctuations, with token confidence recovering from low to high levels. In contrast, incorrect paths can remain stable while following the flawed intermediate context.

These observations do not imply that instability itself is a reliable signal. Recent studies further show that large token fluctuations frequently occur in erroneous paths, making instability alone an ambiguous indicator of correctness (Zhu et al., 2026; Liu et al., 2026). Together, these findings suggest that neither stability nor instability alone can serve as a universal selection criterion. Under standard inference, we examine how the relationship between token stability and correctness varies with the model's confidence in its reasoning path. We require the LLM to verbalize its confidence (Xiong et al., 2023; Yang et al., 2024) and introduce *jitter*, a metric that captures token confidence fluctuations across consecutive reasoning segments. Among paths assigned high verbalized confidence, correct paths tend to exhibit lower jitter than incorrect ones. Stability thus helps distinguish correct paths from errors within a group that the model already considers reliable. By contrast, token stability becomes a less reliable indicator of correctness when the model is uncertain. Retaining unstable paths can preserve potentially valid solutions that would otherwise be discarded solely on the basis of their fluctuations.

Building on this analysis, we introduce the **Verbalized-Token Synergistic Filter** (**VeTo**), a training-free method that combines macro-level verbalized confidence with micro-level token confidence fluctuations. VeTo first uses verbalized confidence to construct the candidate pool and then applies *jitter* according to the resulting confidence regime. It favors stable paths within the high-confidence pool and retains paths with larger fluctuations as a fallback when this pool is empty. The remaining paths are aggregated to produce the final answer.

We evaluate VeTo using six models from the Qwen, Llama, and InternLM families across eight benchmarks spanning mathematical, multidisciplinary, code, and multimodal reasoning. Among all compared strategies, VeTo achieves the highest average accuracy within each model family. We further analyze its behavior across inference budgets from $N = 4$ to 64 and at Qwen model sizes of 4B, 8B, and 14B, finding that the gains generally persist across these settings. In addition, VeTo obtains competitive confidence calibration performance on mathematical reasoning tasks. Overall, our analysis and experiments show how model confidence can guide the use of both stability and instability for reasoning path selection.

2 Related Work

2.1 Efficient Reasoning and Scaling

Within parallel test-time scaling, Self-Consistency (SC) aggregates answers across sampled reasoning paths, whereas Best-of-N (BoN) selects high-scoring candidates (Wang et al., 2022; Brown et al., 2024). Subsequent work improves either sampling efficiency or candidate selection. Early-stopping SC (Li et al., 2024) and difficulty-adaptive SC (Wang et al., 2025) reduce redundant sampling, while Soft-SC (Wang et al., 2024) and DeepConf (Fu et al., 2026) refine aggregation and filtering using likelihood or local token confidence. Our work complements prior work by analyzing both stability and instability as signals for reasoning path selection.

2.2 Uncertainty and Confidence Estimation

LLM confidence can be estimated from internal signals or elicited through self-evaluation. Internal approaches derive confidence from token probabilities, including token-level entropy (Fadeeva et al., 2024) and sequence-level entropy (Sharma & Chopra, 2025). Self-evaluation methods ask the model to assess its own answer through $P(\text{True})$ (Kadavath et al., 2022) or verbalized confidence (Xiong et al., 2023; Yang et al., 2024; Zhou et al., 2023). Specifically, verbalized confidence can be elicited through prompting or learned through fine-tuning (Jang et al., 2025), offering greater interpretability (Band et al., 2024) and supporting iterative prompting (Abbasi Yadkori et al., 2024). Recent studies have also analyzed local uncertainty within reasoning paths, reporting associations between localized entropy spikes or sudden token confidence drops and reasoning errors (Zhu et al., 2026; Liu et al., 2026). In this paper, we examine how token stability relates to correctness across verbalized confidence regimes and combine it with verbalized confidence for reasoning path selection.

3 Stability Analysis and Path Selection

We first analyze the relationship between token stability and correctness across verbalized confidence regimes, using jitter to quantify token confidence fluctuations. Building on this analysis, we introduce VeTo, a training-free method that combines verbalized confidence and jitter to filter reasoning paths.

3.1 Stability and Correctness across Verbalized Confidence Regimes

We use *jitter* to capture token confidence fluctuations, as defined in Section 3.2. To examine how the relationship between token stability and correctness varies with model confidence, we group individual paths by their verbalized confidence. Figure 2 presents representative cases and summarizes token stability and correctness across these confidence regimes under standard inference.

Correct paths generally exhibit higher overall token probabilities than incorrect ones, but the role of stability differs substantially across confidence regimes. Among high-confidence paths, correct solutions tend to have both higher average token confidence and lower jitter, with both signals achieving $\text{AUC} > 0.76$ in distinguishing correct from incorrect paths. Consistent with this pattern, a point-biserial correlation test shows a negative correlation between jitter and correctness ($r = -0.4117$, $p < 0.001$), suggesting that stability provides a useful selection signal when the model is confident. This separation weakens in the low-confidence regime. Although average token confidence remains informative ($\text{AUC} = 0.7094$), higher jitter no longer consistently indicates an incorrect path, and

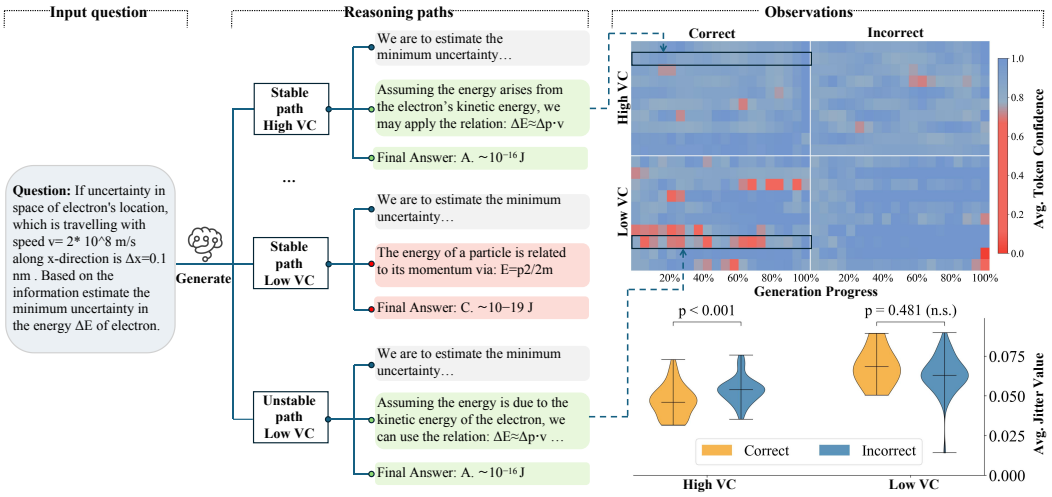


Figure 2: **Token stability and correctness across verbalized confidence regimes.** Representative cases and aggregate statistics from standard inference. Correct paths are more stable on average under high verbalized confidence, while the low-confidence cases include both stable errors and unstable correct paths. Reported p-values use two-sided Mann–Whitney U tests.

unstable paths can yield correct solutions. This shift is also reflected in the jitter gaps between correct and incorrect paths. For example, for Qwen3-8B on AIME25 and Llama-3.1-8B on MATH500, the average gaps (correct minus incorrect) are -0.012 and -0.004 at high confidence, but become $+0.003$ and $+0.009$ at low confidence. These results suggest that stability is informative for selecting among confident paths, whereas unstable paths remain worth considering when the model is uncertain. Appendix Figures 5–7 provide detailed results across models and benchmarks.

3.2 Token Interval Fluctuation Metric

To capture token stability dynamics, we introduce the token interval fluctuation metric (referred to as *jitter*, J). This metric measures changes in mean token confidence between consecutive windows.

For a reasoning path y, let $P = (p_1, p_2, \ldots, p_L)$ denote its token confidence sequence. We partition P into consecutive non-overlapping windows. Each complete window contains W tokens. If the final incomplete window contains at least two tokens, it is retained as an additional window; a single remaining token is discarded. The resulting number of windows is

$$M = \left\lfloor \frac{L}{W} \right\rfloor + \mathbb{I}[L \bmod W \geq 2].$$

Let B_m denote the token indices in the m-th retained window. Its local confidence mean is

$$\mu_m = \frac{1}{|B_m|} \sum_{k \in B_m} p_k. \tag{1}$$

We measure the local fluctuation as the absolute difference between the means of adjacent windows:

$$\delta_m = |\mu_{m+1} - \mu_m|, \quad m = 1, \ldots, M-1. \tag{2}$$

If fewer than two windows are retained ($M < 2$), no adjacent-window difference can be computed, and we set $J(y) = 0$. Otherwise, jitter is the mean absolute difference between adjacent windows:

$$J(y) = \frac{1}{M-1} \sum_{m=1}^{M-1} \delta_m. \tag{3}$$

A larger $J(y)$ indicates greater local variation in token confidence.

3.3 Path Selection via Verbalized Confidence and Token Jitter

Let $\mathcal{C} = \{c_1, \ldots, c_N\}$ denote the sampled reasoning paths. For each path c_i, the model provides a verbalized confidence score $s_i = f_{\mathrm{LLM}}(c_i)$, and we compute its jitter $J_i = J(c_i)$ using Equation 3. We first form the high-confidence pool:

$$\mathcal{C}_{\mathrm{high}} = \{c_i \in \mathcal{C} \mid s_i \geq \tau\},$$

For a controlled evaluation, the main verbalization prompt produces binary scores $s_i \in \{0, 100\}$. This provides a simple confidence partition without introducing an additional cutoff hyperparameter. We therefore fix $\tau = 100$ across all models and benchmarks. Appendix Table 9 additionally evaluates numeric verbalized confidence on a 0–100 scale. The elite pool is then:

$$\mathcal{C}_{\mathrm{elite}} = \begin{cases} \mathcal{C}_{\mathrm{high}}, & \mathcal{C}_{\mathrm{high}} \neq \emptyset, \\ \mathcal{C}, & \text{otherwise.} \end{cases}$$

If $\mathcal{C}_{\mathrm{high}}$ is nonempty, we set θ to the $(100 - \alpha)$-th percentile of the jitter values in $\mathcal{C}_{\mathrm{elite}}$ and retain more stable paths with $J_i \leq \theta$. Otherwise, we set θ to the $(100 - \beta)$-th percentile and retain paths with $J_i \geq \theta$, removing the lowest-fluctuation paths from the fallback pool. This confidence-conditioned rule is motivated by the analysis in Section 3.1. When high-confidence paths are available, lower jitter is used to prioritize stable candidates within a pool that the model considers reliable. When this pool is empty, we use higher-jitter retention as a fallback heuristic, allowing unstable candidates to remain under consideration in the fallback pool. Verbalized confidence therefore determines both the candidate pool and how jitter is interpreted, while α and β control the filtering strength in the two branches. Finally, majority voting over the retained paths $\mathcal{C}_{\mathrm{final}}$ produces the answer $\hat{a}$. In this process, verbalized confidence and jitter are used for candidate selection rather than vote weighting; each retained path contributes equally to the final vote. The complete procedure is summarized in Algorithm 1.

To directly evaluate the fallback heuristic, Appendix Tables 6 and 7 report correct path recall and final voting accuracy, respectively, for instances that enter the fallback branch. In addition, an optional confidence refinement procedure used only for calibration evaluation is described in Appendix Section C.

Algorithm 1 Verbalized-Token Synergistic Filter (VeTo)

Require: Question x, model $\mathcal{M}$, count N, ratios α, β, τ
Ensure: Final answer $\hat{a}$

$\mathcal{C} \leftarrow \emptyset$
for $i = 1$ **to** N **do**
  $(y_i, s_i) \sim \mathcal{M}(x)$
  $J_i \leftarrow \mathrm{CalcJitter}(y_i)$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{(y_i, s_i, J_i)\}$
end for
$\mathcal{C}_{\mathrm{high}} \leftarrow \{c \in \mathcal{C} \mid c.s \geq \tau\}$
if $\mathcal{C}_{\mathrm{high}} \neq \emptyset$ **then**
  $\mathcal{C}_{\mathrm{elite}} \leftarrow \mathcal{C}_{\mathrm{high}}$
  $\theta \leftarrow \mathrm{Percentile}(\{c.J : c \in \mathcal{C}_{\mathrm{elite}}\}, 100 - \alpha)$
  $\mathcal{C}_{\mathrm{final}} \leftarrow \{c \in \mathcal{C}_{\mathrm{elite}} \mid c.J \leq \theta\}$
else
  $\mathcal{C}_{\mathrm{elite}} \leftarrow \mathcal{C}$
  $\theta \leftarrow \mathrm{Percentile}(\{c.J : c \in \mathcal{C}_{\mathrm{elite}}\}, 100 - \beta)$
  $\mathcal{C}_{\mathrm{final}} \leftarrow \{c \in \mathcal{C}_{\mathrm{elite}} \mid c.J \geq \theta\}$
end if
$\hat{a} \leftarrow \mathrm{MajorityVote}(\{\mathrm{Extract}(c.y) : c \in \mathcal{C}_{\mathrm{final}}\})$
return $\hat{a}$

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models. We evaluate six models from three families: Qwen3-8B (Yang et al., 2025) and Qwen3-VL-8B (Bai et al., 2025), Llama-3.1-8B and Llama-3.2-11B-Vision (Grattafiori et al., 2024), and InternLM3-8B (InternLM Team, 2025) and InternVL3-8B (Zhu et al., 2025). They cover both language-only and vision-language architectures.

Benchmarks. We evaluate on eight benchmarks spanning four reasoning settings. GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), and AIME25 (Zhang & Math-AI, 2025) cover mathematical reasoning; GPQA-Diamond (Rein et al., 2024) and R-Bench (Guo et al., 2025) cover advanced multidisciplinary reasoning; MBPP+ (Liu et al., 2023) evaluates code generation; and MathVista (Lu et al., 2023) and MMMU-Math (Yue et al., 2024) evaluate multimodal mathematical reasoning. Further details are provided in Appendix Section B.

Baselines. We compare VeTo with consensus-driven methods and methods based on token confidence. The former includes Self-Consistency (Wang et al., 2022), Confidence-Informed SC (Taubenfeld et al., 2025), and Complexity-Weighted SC (Fu et al., 2022). The latter includes Average Trace Confidence (Kang et al., 2025; Fu et al., 2026) and Average Tail Confidence (Fu et al., 2026). We also report the base model performance.

Experimental Settings. We randomly sample 250 instances from the test sets of large-scale benchmarks (e.g., GSM8K and MATH500) and use the full test sets for smaller benchmarks (e.g., GPQA-Diamond and AIME25). Unless otherwise specified, we sample $N = 16$ reasoning paths per instance with temperature 0.7 and top-p 0.95. Token-based baseline filtering strategies retain paths according to the η-th percentile of their metric scores. We use two jitter window sizes per model family: one shared across all text benchmarks, including code, and one across both multimodal benchmarks. Configurations, selection details, and sensitivity analysis are provided in Appendix Section E.2. To examine different filtering strengths, we evaluate three variants: Conservative ($\alpha = 10$, $\beta = 90$; $\eta = 10$), Asymmetric ($\alpha = 10$, $\beta = 10$), and Aggressive ($\alpha = 90$, $\beta = 10$; $\eta = 90$). The Conservative variant applies moderate filtering in both regimes, favoring stable paths in the high-confidence pool while preserving potentially useful unstable paths in the fallback pool. The Asymmetric variant uses the same high-confidence filtering as Conservative but applies stricter filtering in the fallback branch. The Aggressive variant applies strong filtering in both regimes. Threshold ablations are provided in Appendix Tables 11 and 12.

4.2 VETO IMPROVES TEXT REASONING

As shown in Table 1, VeTo maintains GSM8K performance while yielding larger gains on GPQA-Diamond and R-Bench. On GSM8K, models are generally confident: approximately 1.4% of correct Qwen3-8B paths have low verbalized confidence. VeTo therefore favors stable paths within the high-confidence pool, achieving 88.0% accuracy. On GPQA-Diamond and R-Bench, models are more often uncertain; Qwen3-8B assigns low verbalized confidence to nearly half of its correct GPQA-Diamond paths. In such cases, SC can be dominated by consistently generated errors. By contrast, when the high-confidence pool is empty, VeTo retains higher-jitter paths that can yield correct solutions. Overall, Qwen (Conservative) and Llama (Aggressive) improve over SC by approximately 4.5% and 5.0%, respectively, on GPQA-Diamond.

Overall, the Conservative variant achieves the highest average accuracy for Qwen and Llama, while the Asymmetric variant performs best for InternLM. The Aggressive variant filters strictly across both regimes, but heavily penalizing instability in high-confidence states introduces more risk than reward. Verbalized confidence therefore helps identify a more reliable candidate pool. Indeed, high-confidence Qwen3-8B paths achieve 87.7% accuracy on GSM8K; on R-Bench, they reach 25.0%, compared with 11.2% for low-confidence paths. However, verbalized confidence alone cannot distinguish correct from incorrect paths within the high-confidence pool. Within this pool, aggressive filtering risks discarding correct paths along with incorrect ones; the Asymmetric and Conservative variants limit this risk by moderating filtering. Moreover, on GSM8K, the Asymmetric variant of Llama-3.1-8B underperforms SC, consistent with the model's lack of confidence in its reasoning paths.

Table 1: **Performance comparison of VeTo against baseline methods across model families.** Accuracy (%) is reported. Baselines comprise consensus-driven methods (SC, CISC, CW-SC) and methods based on token confidence (AVG_{Trace}, AVG_{Tail}). Asterisks mark multimodal benchmarks; dashes denote unavailable results; **bold** marks the best average within each model family. Two-sided exact McNemar tests on pooled VeTo (Conservative) and SC outcomes yield $p = 1.13 \times 10^{-5}$ for Qwen, $p = 0.0015$ for Llama, and $p = 0.0567$ for InternLM.

Method	GSM8K	MATH500	AIME25	GPQA-Diamond	R-Bench	MBPP+	MathVista*	MMMU-Math*	Average
Qwen3-8B & Qwen3-VL-8B*									
Base Model	84.7	82.6	18.8	41.6	17.3	66.0	66.3	55.3	54.1
SC	87.6	86.8	26.7	45.5	15.6	66.0	69.2	61.2	57.3
CISC	88.0	88.4	26.7	45.5	17.2	67.2	69.6	62.4	58.1
CW-SC	88.0	87.2	23.3	48.0	15.6	65.6	69.2	60.4	57.2
AVG_{Trace} (Conservative)	87.6	87.2	30.0	45.5	15.2	65.6	68.8	61.6	57.7
AVG_{Trace} (Aggressive)	86.4	85.2	16.7	39.9	16.0	67.2	66.8	63.2	55.2
AVG_{Tail} (Conservative)	87.2	87.2	30.0	45.5	15.6	66.0	68.8	62.8	57.9
AVG_{Tail} (Aggressive)	85.2	84.0	13.3	43.9	15.6	63.6	67.2	59.6	54.1
VeTo (Conservative)	88.0	90.0	33.3	50.0	19.2	68.0	69.6	63.2	**60.2**
VeTo (Asymmetric)	88.0	90.0	33.3	45.5	20.0	68.0	69.6	63.2	59.7
VeTo (Aggressive)	86.8	86.4	20.0	44.9	20.0	64.4	68.0	61.2	56.5
Llama-3.1-8B & Llama-3.2-11B-Vision*									
Base Model	83.8	53.7	-	26.5	39.1	56.0	34.1	31.4	46.4
SC	91.2	68.4	-	29.3	45.6	56.0	39.6	36.8	52.4
CISC	90.0	58.8	-	30.8	49.2	58.0	32.8	40.8	51.5
CW-SC	89.6	56.4	-	23.7	32.0	53.6	40.0	33.2	46.9
AVG_{Trace} (Conservative)	91.2	67.6	-	28.3	44.0	56.4	39.2	38.4	52.2
AVG_{Trace} (Aggressive)	86.4	52.0	-	21.2	27.6	57.6	35.2	26.0	43.7
AVG_{Tail} (Conservative)	91.2	68.4	-	29.8	45.6	56.0	39.6	37.6	52.6
AVG_{Tail} (Aggressive)	83.2	47.6	-	20.2	30.8	56.0	34.8	25.2	42.5
VeTo (Conservative)	91.2	69.6	-	33.8	50.0	56.8	38.8	41.2	**54.5**
VeTo (Asymmetric)	87.6	53.2	-	35.4	51.2	56.8	33.6	41.2	51.3
VeTo (Aggressive)	85.6	53.6	-	34.3	48.4	56.8	33.6	36.0	49.8
InternLM3-8B & InternVL3-8B*									
Base Model	91.6	75.4	7.9	38.9	45.0	59.2	50.9	42.6	51.4
SC	93.6	84.0	16.7	39.4	56.0	59.2	57.2	46.4	56.6
CISC	93.6	84.8	16.7	40.9	56.4	59.2	54.4	49.2	56.9
CW-SC	94.0	84.0	13.3	41.4	54.8	53.6	56.0	48.0	55.6
AVG_{Trace} (Conservative)	93.2	84.4	16.7	40.4	55.6	60.0	56.8	46.4	56.7
AVG_{Trace} (Aggressive)	90.8	76.8	3.3	42.4	38.4	57.6	50.4	38.8	49.8
AVG_{Tail} (Conservative)	93.2	84.4	16.7	41.9	54.8	58.0	56.8	46.8	56.6
AVG_{Tail} (Aggressive)	92.4	77.2	3.3	38.9	47.2	61.2	52.0	36.8	51.1
VeTo (Conservative)	94.4	84.4	16.7	42.9	59.2	59.6	53.6	49.6	57.5
VeTo (Asymmetric)	94.4	84.4	16.7	42.9	57.6	59.6	53.2	53.2	**57.7**
VeTo (Aggressive)	92.4	79.6	13.3	42.4	49.6	59.2	51.6	49.2	54.7

4.3 VeTo on Multimodal Reasoning

As shown in Table 1, the Asymmetric variant on Qwen3-VL-8B achieves 69.6% accuracy on MathVista and 63.2% on MMMU-Math, outperforming SC on both benchmarks. For InternVL3-8B, the Asymmetric variant improves MMMU-Math accuracy from 46.4% to 53.2%, although all three variants underperform SC on MathVista (51.6–53.6% vs. 57.2%).

The Llama results further reveal a practical limitation of relying on verbalized confidence to guide path selection: when confidence scores are poorly informative, they can restrict how effectively stability and instability signals are used. On MathVista, VeTo achieves 38.8% accuracy with Llama-3.2-11B-Vision, slightly below the SC baseline of 39.6%, while CISC drops further to 32.8%. This behavior is associated with extreme underconfidence: the model assigns low verbalized confidence to 80.7% of its correctly generated paths, limiting the ability of verbalized confidence to identify a reliable candidate pool. VeTo partially mitigates this issue by additionally using token confidence fluctuations to retain some of these paths.

On MMMU-Math, the model instead exhibits a common pattern among LLMs: overconfidence, assigning high verbalized confidence even to incorrect paths. Within the high-confidence pool, the Conservative variant can still exploit token-level stability and improves over SC from 36.8% to 41.2%.

4.4 QWEN EVALUATOR FOR LLAMA

Motivated by the confidence patterns observed for Llama, we retain Llama as the path generator but use the corresponding Qwen3 model as the confidence evaluator. The generated paths and window sizes remain unchanged, allowing us to examine how the confidence evaluator affects selection from the same candidate pool. Compared with the Llama-evaluated results in Table 1, the Qwen evaluator improves the Conservative variant on five of the seven available benchmarks and raises its average accuracy from 54.5% to 55.3%. The largest improvement occurs on MATH500, where accuracy increases from 69.6% to 72.4%. The Asymmetric and Aggressive variants show mixed changes relative to their Llama-evaluated counterparts. These results indicate that evaluator choice can improve path selection, although its benefits depend on the filtering configuration.

Table 2: **Qwen evaluator for Llama-generated paths.** Accuracy (%) is reported. Qwen3-8B serves as the evaluator for Llama-3.1-8B on text benchmarks, while Qwen3-VL-8B evaluates Llama-3.2-11B-Vision on multimodal benchmarks. Asterisks mark multimodal benchmarks; dashes denote unavailable results; **bold** marks the best average.

Method	GSM8K	MATH500	AIME25	GPQA-Diamond	R-Bench	MBPP+	MathVista*	MMMU-Math*	Average
SC	91.2	68.4	-	29.3	45.6	56.0	39.6	36.8	52.4
VeTo w/ Qwen (Conservative)	92.4	72.4	-	34.8	48.4	58.4	40.0	40.8	**55.3**
VeTo w/ Qwen (Asymmetric)	91.6	63.2	-	29.8	45.6	58.4	40.0	40.8	52.8
VeTo w/ Qwen (Aggressive)	90.4	61.2	-	29.3	45.6	60.8	36.4	35.2	51.3

4.5 ABLATION OF VERBALIZED CONFIDENCE AND TOKEN JITTER

We compare two ablations across all eight benchmarks for the Qwen models under the main experimental settings, using Qwen3-8B and Qwen3-VL-8B. VeTo w/o Jitter uses verbalized confidence pooling without further token-level filtering, whereas VeTo w/o Verbalized Confidence applies low-jitter filtering to the full candidate pool with $\alpha = 10$.

Table 3: **Component ablation across all Qwen family benchmarks.** Accuracy (%) is reported. Asterisks mark multimodal benchmarks; **bold** marks the best average.

Method	GSM8K	MATH500	AIME25	GPQA-D.	R-Bench	MBPP+	MathVista*	MMMU-Math*	Average
SC	87.6	86.8	26.7	45.5	15.6	66.0	69.2	61.2	57.3
VeTo w/o Jitter	88.0	88.4	26.7	47.5	16.8	67.2	70.0	61.2	58.2
VeTo w/o Verbalized Confidence	87.6	87.6	33.3	43.9	16.0	66.8	68.8	61.2	58.2
VeTo (Conservative)	88.0	90.0	33.3	50.0	19.2	68.0	69.6	63.2	**60.2**

VeTo (Conservative) achieves the highest macro-average accuracy at 60.2%, with absolute gains of 2.85%, 1.95%, and 2.01% over SC, VeTo w/o Jitter, and VeTo w/o Verbalized Confidence, respectively. It is also best or tied for best on seven of the eight benchmarks. Compared with removing jitter, removing verbalized confidence causes larger drops on GSM8K, MATH500, GPQA-Diamond, R-Bench, MBPP+, and MathVista. Removing jitter lowers average accuracy, although VC pooling alone performs better on MathVista. These results indicate that verbalized confidence and token-level stability contribute complementary information to path selection.

To isolate the token-selection rule, Appendix Table 8 compares jitter with average token confidence and random pruning under identical confidence pools and matched retention counts. On the evaluated Qwen candidate pools, VeTo improves average accuracy by 1.545% and 1.820%, respectively, with 95% bootstrap percentile confidence intervals of [0.692, 2.653]% and [0.940, 2.877]%.

4.6 SCALABILITY ACROSS INFERENCE BUDGETS AND MODEL SIZES

Figure 3 examines scalability across inference budgets from $N = 4$ to 64. Across text and multimodal tasks, the Conservative variant outperforms SC in most evaluated Qwen settings, although gains vary with the budget. On MMMU-Math with Qwen3-VL-8B, SC accuracy decreases from $N = 32$ to 64, whereas all three VeTo variants improve, illustrating the value of path selection when additional samples no longer benefit unfiltered majority voting. The trend is less consistent for Llama, whose curves flatten or fluctuate across benchmarks and filtering configurations.

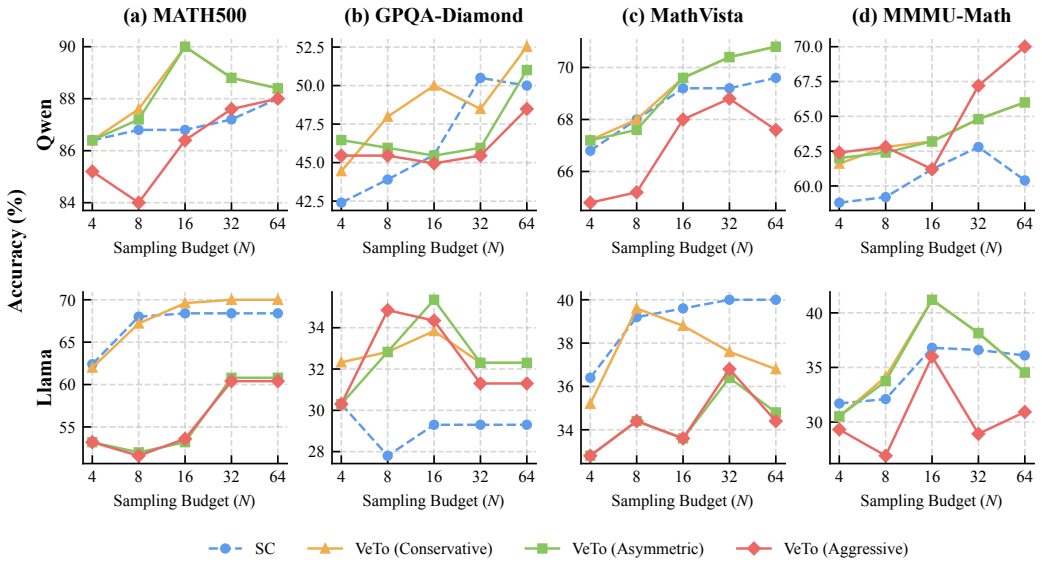


Figure 3: **Accuracy scaling across inference budgets** ($N = 4$–64)**.** Top: Qwen3-8B **(a, b)** and Qwen3-VL-8B **(c, d)**. Bottom: Llama-3.1-8B **(a, b)** and Llama-3.2-11B-Vision **(c, d)**.

Figure 4 further examines model-size scaling for Qwen3-4B and Qwen3-14B. At 4B, the Conservative and Asymmetric variants outperform SC across all three benchmarks. At 14B, both match SC on GSM8K and improve on GPQA-Diamond, while Conservative also improves AIME25 accuracy. The Qwen3-8B results in Table 1 add the intermediate scale, covering 4B, 8B, and 14B. Although the gains are not strictly monotonic with model size, they generally persist across all three scales.

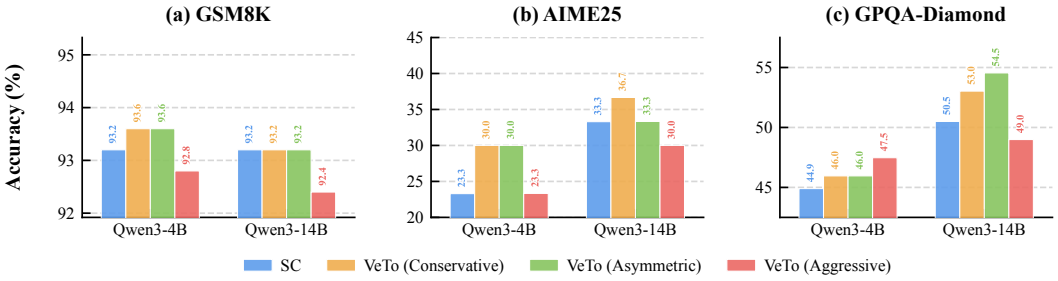


Figure 4: **Accuracy scaling across model sizes.** SC and the three VeTo variants on Qwen3-4B and Qwen3-14B across **(a)** GSM8K, **(b)** AIME25, and **(c)** GPQA-Diamond.

5 Conclusion

This work analyzes how the relationship between token stability and correctness varies with the model's verbalized confidence. Based on this analysis, we introduce VeTo, a training-free method that combines verbalized confidence with token confidence fluctuations. VeTo uses verbalized confidence to guide how stable and unstable paths are treated during selection. Across eight benchmarks and three model families, VeTo achieves the highest average accuracy within each family among the compared strategies. Ablation studies further show that both verbalized confidence and jitter contribute to performance, while scaling experiments indicate that VeTo's gains persist across inference budgets and model sizes. Our findings show that stability and instability can serve as complementary signals for reasoning path selection when interpreted in conjunction with verbalized confidence.

AI USE STATEMENT

The authors used AI tools to improve language clarity and readability and to assist with code development and debugging. All AI-assisted content and code remain subject to their review and verification, and they take full responsibility for the accuracy and integrity of this work.

REPRODUCIBILITY STATEMENT

The authors provide detailed descriptions of the proposed method and experimental settings to facilitate reproduction and verification. Section 3 defines the token fluctuation metric and filtering procedure, with pseudocode in Algorithm 1. The experimental setup is described in the main text. Additional dataset details, hyperparameter settings, and verbalized confidence prompts are provided in the appendices. The experiments use publicly available models and benchmarks, including GSM8K, MATH500, and MathVista, to support independent evaluation.

REFERENCES

Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvari. To believe or not to believe your llm: Iterative prompting for estimating epistemic uncertainty. *Advances in Neural Information Processing Systems*, 37:58077–58117, 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. Linguistic calibration of long-form generations. *arXiv preprint arXiv:2404.00474*, 2024.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

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

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, et al. Fact-checking the output of large language models via token-level uncertainty quantification. *arXiv preprint arXiv:2403.04696*, 2024.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*, 2022.

Yichao Fu, Xuewei Wang, Hao Zhang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=8LqHs0KIM7.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Meng-Hao Guo, Jiajun Xu, Yi Zhang, Jiaxi Song, Haoyang Peng, Yi-Xuan Deng, Xinzhi Dong, Kiyohiro Nakayama, Zhengyang Geng, Chen Wang, et al. R-Bench: Graduate-level multi-disciplinary benchmarks for LLM & MLLM complex reasoning evaluation. *arXiv preprint arXiv:2505.02018*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

InternLM Team. InternLM3-8B-Instruct. https://huggingface.co/internlm/internlm3-8b-instruct, 2025. Official model card.

Chaeyun Jang, Moonseok Choi, Yegon Kim, Hyungi Lee, and Juho Lee. Verbalized confidence triggers self-verification: Emergent behavior without explicit reasoning supervision. *arXiv preprint arXiv:2506.03723*, 2025.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. *arXiv preprint arXiv:2502.18581*, 2025.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. *arXiv preprint arXiv:2401.10480*, 2024.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36:21558–21572, 2023.

Xiaoyang Liu, Dawei Wang, Tian Li, Huizhi Liang, Gary Ushaw, and Richard Davison. Tracing the light of thought: A probabilistic self- and cross-consistency verification mechanism improving mathematical reasoning in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 22444–22459, 2026.

Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of AI research. *Nature*, 651(8107):914–919, 2026.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 30th Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

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. In *First Conference on Language Modeling*, 2024.

Aman Sharma and Paras Chopra. Think just enough: Sequence-level entropy as a confidence signal for llm reasoning. *arXiv preprint arXiv:2510.08146*, 2025.

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

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in llms. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20090–20111, 2025.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Soft self-consistency improves language model agents. *arXiv preprint arXiv:2402.13212*, 2024.

Xinglin Wang, Shaoxiong Feng, Yiwei Li, Peiwen Yuan, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. Make every penny count: Difficulty-adaptive self-consistency for cost-efficient reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6919–6932, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*, 2024.

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. *arXiv preprint arXiv:2306.13063*, 2023.

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. *arXiv preprint arXiv:2505.09388*, 2025.

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. On verbalized confidence scores for LLMs. *arXiv preprint arXiv:2412.14737*, 2024.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025. URL `https://huggingface.co/datasets/math-ai/aime25`.

Weike Zhao, Chaoyi Wu, Yanjie Fan, Pengcheng Qiu, Xiaoman Zhang, Yuze Sun, Xiao Zhou, Shuju Zhang, Yu Peng, Yanfeng Wang, et al. An agentic system for rare disease diagnosis with traceable reasoning. *Nature*, 651(8106):775–784, 2026.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori B Hashimoto. Navigating the grey area: How expressions of uncertainty and overconfidence affect language models. In *Proceedings of the 28th Conference on Empirical Methods in Natural Language Processing*, pp. 5506–5524, 2023.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Wei Zhu, Jian Zhang, Lixing Yu, Kun Yue, and Zhiwen Tang. Dissecting failure dynamics in large language model reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8893–8914, 2026.

A Additional Analysis

A.1 Controlled Misleading Prefix Construction

To reduce confounding from problems the models cannot solve, we construct controlled GSM8K and MATH500 datasets from reasoning paths with correct answers. We retain each path's initial portion, append a deliberately erroneous step, and truncate the remaining solution. The model then continues from this misleading prefix. For each model, we construct 200 examples in total, distributed approximately evenly across the following three error types:

- **Logic Error:** Misinterpret relationships between quantities and apply an incorrect operation, such as addition instead of subtraction or division instead of multiplication.
- **Variable Misattribution:** Use a value associated with the wrong variable or entity in a reasoning step.
- **Goal Misalignment:** Introduce reasoning steps that pursue a different objective from the one specified in the question.

Verbalized confidence on controlled errors. After continuation, we prompt the model to assess its answer confidence. Table 4 reports normalized scores for correct and incorrect paths.

Table 4: **Verbalized confidence under misleading prefixes.** Average normalized scores ($[0, 1]$) on GSM8K and MATH500 are grouped by final answer correctness after continuation.

	GSM8K		MATH500	
Error Type	**Correct**	**Incorrect**	**Correct**	**Incorrect**
Qwen3-8B				
Logic Error	0.90	0.77	1.00	0.65
Variable Misattribution	1.00	0.88	1.00	0.92
Goal Misalignment	0.98	0.53	1.00	0.92
Llama-3.1-8B				
Logic Error	0.62	0.65	0.78	0.62
Variable Misattribution	0.71	0.67	0.76	0.64
Goal Misalignment	0.59	0.62	0.66	0.58

Qwen3-8B assigns higher average confidence to correct paths across all six dataset–error combinations, but incorrect paths also receive substantial confidence, reaching 0.92 on MATH500. For Llama-3.1-8B, the separation is less consistent: on GSM8K, incorrect paths receive slightly higher average confidence for Goal Misalignment and Logic Error.

The boxed examples below illustrate the three error types constructed from GSM8K, with Qwen3-8B continuing from each misleading prefix. The injected steps are highlighted in red.

Logic Error: Example 1

Question: John takes care of 10 dogs. Each dog takes 0.5 hours a day to walk and take care of their business. How many hours a week does he spend taking care of dogs?
Ground-truth Answer: 35 **Error Type:** Logic Error
Erroneous Prefix:
Step 1: Calculate the daily time spent on one dog.
Each dog takes 0.5 hours per day to walk and take care of their business.
Step 2: Multiply the daily time per dog by the number of dogs.
John takes care of 10 dogs, so:
$10 \text{ dogs} \div 0.5 \text{ hours/dog} = 20 \text{ hours/day}$
Continuation Text:
Step 3: Multiply the daily time by the number of days in a week.
There are 7 days in a week, so:
$20 \text{ hours/day} \times 7 \text{ days/week} = 140 \text{ hours/week}$.
140

Logic Error: Example 2

Question: Isabelle bought party supplies for her little sister's birthday party. She spent $12 on ingredients for the cake, $43 on birthday presents, $15 on decorations, $4 on invitations, and $22 on goodie bags for the party guests. She split the cost evenly three ways with her two parents. How many dollars did each person pay?
Ground-truth Answer: 32 **Error Type:** Logic Error
Erroneous Prefix:
Step 1: **Calculate the total cost of all the party supplies.**
Add up the amounts spent on each item:
- $12 (cake ingredients)
- $43 (birthday presents)
- $15 (decorations)
- $4 (invitations)
- $22 (goodie bags)

$$12 + 43 + 15 + 4 + 22 = 96$$

Step 2: **Determine how many people are splitting the cost.**
Isabelle split the cost evenly with her two parents, so that's a total of **3 people**.
Step 3: **Divide the total cost by the number of people.**

$$96 \times 3 = 288$$

Continuation Text:

$$288 \div 3 = 96$$

Each person paid $96.

Variable Misattribution: Example 1

Question: Adrian's age is three times the age of Harriet, and Harriet is half the age of Zack. Calculate the average age of the three in three years if Harriet is 21 years old now.
Ground-truth Answer: 45 **Error Type:** Variable Misattribution
Erroneous Prefix:
Step 1: Identify the current ages of Adrian, Harriet, and Zack.
We are given that Harriet is 21 years old now.
- Harriet's age = 21 years
Step 2: Use the relationship between Adrian's age and Harriet's age.
Adrian's age is three times Harriet's age:
- Adrian's age = $3\times$ Zack's age = $3 \times 42 = 126$ years
Continuation Text:
Step 3: Use the relationship between Harriet's age and Zack's age.
Harriet is half the age of Zack:
- Zack's age = $2\times$ Harriet's age = $2 \times 21 = 42$ years
Step 4: Calculate the current total age of the three.
- Total current age = Adrian's age + Harriet's age + Zack's age
- Total current age = $126 + 21 + 42 = 189$ years
Step 5: Calculate the total age in three years.
- Each person will be 3 years older, so total increase = $3 \times 3 = 9$ years
- Total age in three years = $189 + 9 = 198$ years
Step 6: Calculate the average age in three years.
- Average age = Total age in three years $\div$ 3
- Average age = $198 \div 3 = 66$ years
66

Variable Misattribution: Example 2

Question: Mrs. Cruz is looking for a house that will not go beyond her \$400 000 budget. She saw a property that has a selling price of \$350 000. On top of that, the buyer has to pay a brokerage fee which is 5% of the selling price, and also the transfer fee that is 12% of the selling price. How much more is the total price of the house than Mrs. Cruz's budget?
Ground-truth Answer: 9500 **Error Type:** Variable Misattribution
Erroneous Prefix:
Step 1: Calculate the brokerage fee.
Brokerage fee = 5% of \$350,000
$= 0.05 \times 350{,}000$
= \$17,500
Step 2: Calculate the transfer fee.
Transfer fee = 12% of \$400,000
$= 0.12 \times 400{,}000$
= \$48,000
Continuation Text:
Step 3: Calculate the total cost of the house.
Total cost = Selling price + Brokerage fee + Transfer fee
= \$350,000 + \$17,500 + \$48,000
= \$415,500
Step 4: Calculate how much more the total cost is than Mrs. Cruz's budget.
Difference = Total cost $-$ Budget
= \$415,500 $-$ \$400,000
= \$15,500
\$15,500

Goal Misalignment: Example 1

Question: Rosie can run 10 miles per hour for 3 hours. After that, she runs 5 miles per hour. How many miles can she run in 7 hours?
Ground-truth Answer: 50 **Error Type:** Goal Misalignment
Erroneous Prefix:
Step 1: Calculate the distance Rosie runs in the first 3 hours.
She runs at a speed of 10 miles per hour for 3 hours.
Distance = Speed $\times$ Time
Distance = 10 miles/hour $\times$ 3 hours = 30 miles
Step 2: Calculate the remaining time after the first 3 hours.
Total time = 7 hours
Time remaining = 7 hours $-$ 3 hours = 4 hours
Step 3: Calculate the distance Rosie runs in the remaining 4 hours.
She runs at a speed of 5 miles per hour for 4 hours.
Distance = Speed $\times$ Time
Distance = 5 miles/hour $\times$ 4 hours = 20 miles
Step 4: Calculate the average distance per hour for the entire run.
Average distance per hour = (30 miles + 20 miles) $\div$ 7 hours = 50 miles $\div$ 7 hours $\approx$ 7.14 miles/hour
Continuation Text:
Therefore, Rosie can run approximately 7.14 miles in 7 hours.
The answer is approximately 7.14 miles.

Goal Misalignment: Example 2

Question: Sam sells bread. He has a target of selling 120 crates of bread in a week. One week he was closed on Monday and Friday. Over the weekend he sold 20 crates. On Tuesday he sold 15 crates, on Wednesday 12 crates, and Thursday 18 crates. By how many crates was Sam off from his target for the week?
Ground-truth Answer: 55 **Error Type:** Goal Misalignment
Erroneous Prefix (excerpt):
[...]
Step 3: Add up the crates sold on the days he was open. [...]
Total crates sold = $15 + 12 + 18 + 20 = \mathbf{65}$ crates
Step 4: Calculate how many crates Sam was off from his target. [...]
Difference = $120 - 65 = \mathbf{55}$ crates
Now, we must calculate the average crates sold per day.
Continuation Text (excerpt):
Since he was open for 4 days (Tuesday, Wednesday, Thursday, and Saturday), the average is $65/4 = 16.25$ crates per day.
Step 5: Calculate the total number of crates he should have sold if he had been open all 7 days. [...]
Total crates = $16.25 \times 7 = \mathbf{113.75}$ crates
Step 6: Calculate how many crates Sam was off from his target. [...]
Difference = $120 - 113.75 = \mathbf{6.25}$ crates
6.25 crates

A.2 Additional Token Confidence Dynamics

Across Figures 5–7, the top and bottom rows correspond to paths with high and low verbalized confidence, while the columns show average token confidence distributions, token confidence trajectories, and jitter, respectively. Green and red denote correct and incorrect paths. In the high-confidence regime, average jitter is slightly higher for incorrect paths on AIME25 (0.073 vs. 0.061), MATH500 (0.071 vs. 0.067), and MathVista (0.035 vs. 0.033). This pattern is consistent with stability providing an additional selection signal among paths that the model already rates as confident. In the low-confidence regime, this separation disappears and the average ordering slightly reverses: correct paths have average jitter of 0.087 vs. 0.084 on AIME25, 0.073 vs. 0.064 on MATH500, and 0.052 vs. 0.046 on MathVista. Beyond these aggregate values, the left column shows that the relationship between average token confidence and correctness varies across model–benchmark pairs, while the middle column illustrates corresponding differences in token confidence trajectories. Low-confidence correct paths exhibit pronounced local fluctuations on AIME25 and MathVista, whereas the corresponding MATH500 trajectories are more moderate even though their average jitter remains higher than that of incorrect paths. On AIME25, low-confidence correct paths remain sharply concentrated near high average token confidence, while incorrect paths cover a broader range. The MATH500 and MathVista distributions exhibit different overlap patterns, further illustrating why average token confidence or stability alone does not consistently separate correct from incorrect paths in this regime.

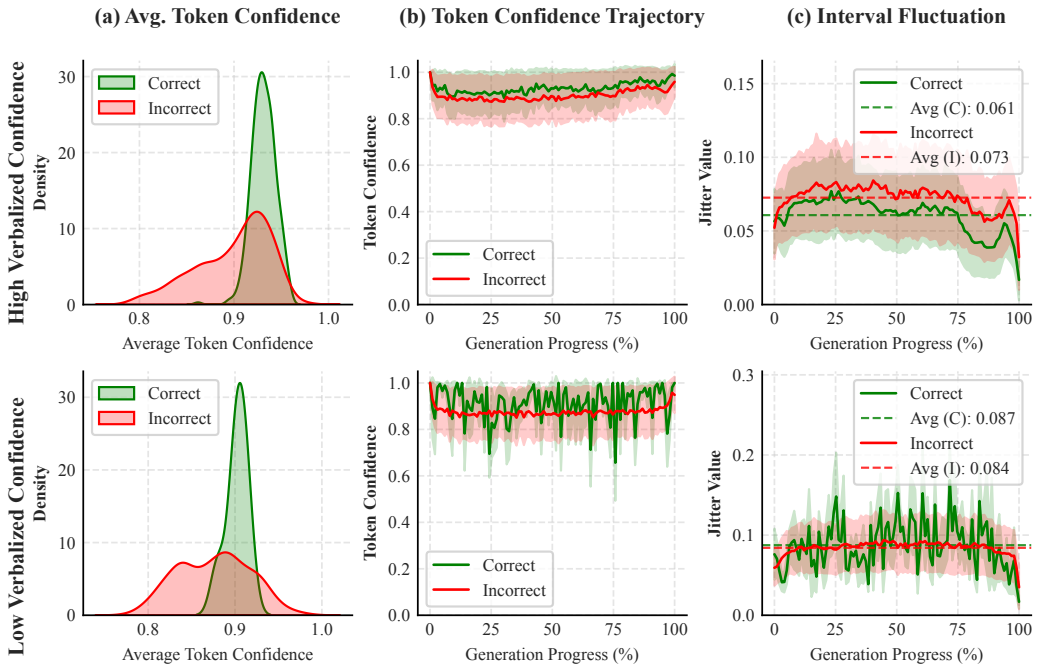


Figure 5: **Token confidence distributions, trajectories, and jitter.** Results for Qwen3-8B on AIME25 under high and low verbalized confidence.

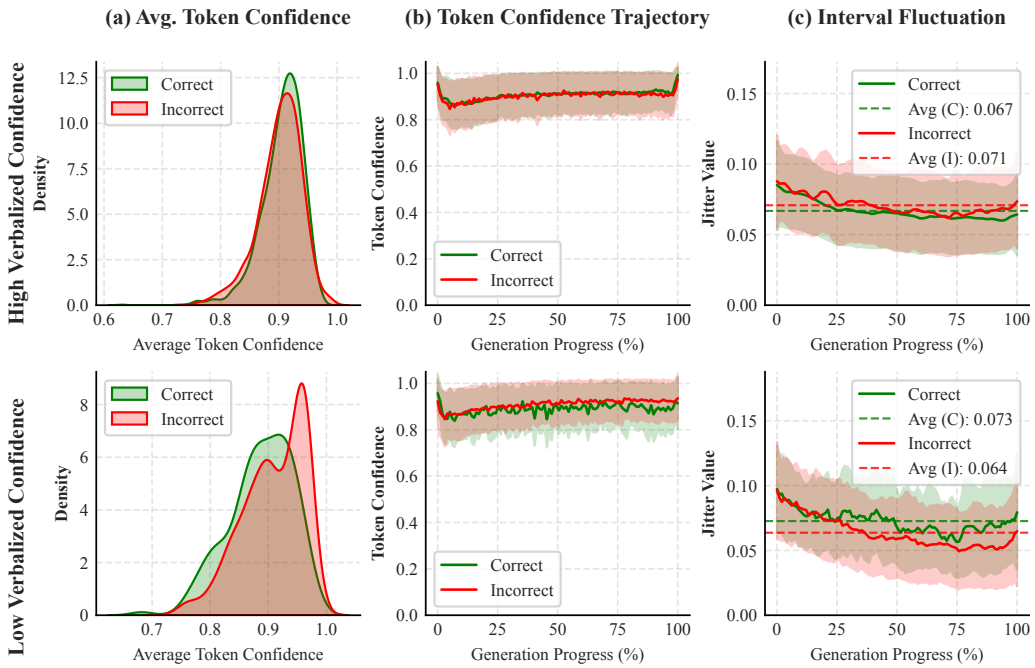


Figure 6: **Token confidence distributions, trajectories, and jitter.** Results for Llama-3.1-8B on MATH500 under high and low verbalized confidence.

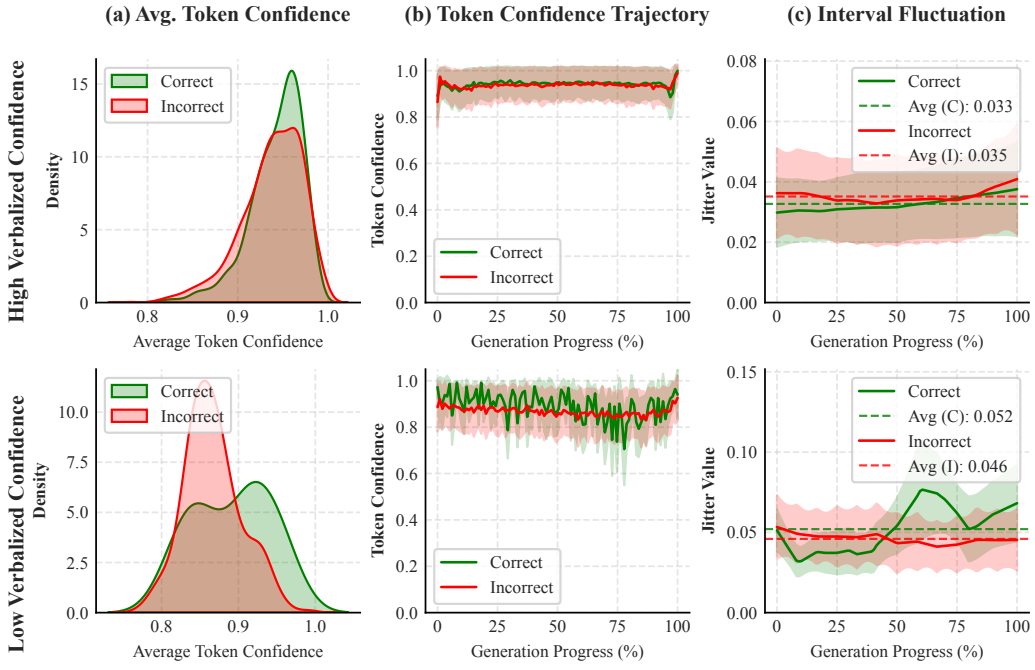


Figure 7: **Token confidence distributions, trajectories, and jitter.** Results for Qwen3-VL-8B on MathVista under high and low verbalized confidence.

A.3 Strategy Analysis

The three VeTo variants balance error removal against the preservation of valid candidates. When the high-confidence pool contains many correct paths, a small α limits accidental deletion while applying a mild preference for stability. If incorrect paths also receive high confidence, jitter can provide an additional selection signal, but increasing α may remove correct paths as well. Table 11 reflects this trade-off: moderate filtering often improves over $\alpha = 0$, whereas aggressive filtering can reduce accuracy.

The fallback branch is triggered only when no sampled path receives high confidence. A large β preserves most candidates, including unstable paths that may yield valid solutions. Reducing β concentrates voting on a smaller high-jitter subset, but also increases the risk of losing correct candidates. Table 12 shows that the accuracy effect varies across model families and task categories, while Table 6 separately measures correct path recall. These trade-offs motivate the three variants: Conservative combines mild high-confidence filtering with broad fallback retention, Asymmetric strengthens only fallback filtering, and Aggressive strengthens both. In Table 1, Conservative has the highest average for Qwen and Llama, and Asymmetric for InternLM. No variant is uniformly best across benchmarks.

A.4 Risks of Verbalized Confidence Pooling

Verbalized confidence determines both which paths enter the candidate pool and how jitter filtering is applied. An incorrect path assigned high confidence is included in the high-confidence pool. Unlike SC, which uses all sampled paths, VeTo restricts filtering to the high-confidence pool whenever $\mathcal{C}_{\text{high}} \neq \emptyset$. Correct paths assigned lower confidence may therefore be excluded before jitter filtering.

However, assigning low confidence to a correct path does not always exclude it. If $\mathcal{C}_{\text{high}}$ is empty, the path remains in the fallback pool, although it may still be removed by the jitter threshold. If the high-confidence pool is nonempty, the path is excluded before jitter filtering. The effects of these confidence errors therefore depend on the other candidates and the subsequent filtering step. The Qwen evaluator experiment in Table 2 examines this sensitivity while keeping the generated paths unchanged. This distinction between pool selection and within-pool filtering helps explain why evaluator quality and token stability both affect the final decision.

B Dataset Details

Our experiments cover eight benchmarks across mathematical, multidisciplinary, code, and multimodal reasoning:

- **GSM8K** (Cobbe et al., 2021) is a dataset of grade-school mathematical word problems that require multi-step reasoning.
- **MATH500** (Hendrycks et al., 2021) contains 500 high-school competition mathematics problems drawn from seven subject areas, including algebra, geometry, and number theory.
- **AIME25** (Zhang & Math-AI, 2025) consists of problems from the 2025 American Invitational Mathematics Examination and evaluates competition-level mathematical reasoning.
- **GPQA-Diamond** (Rein et al., 2024) is the most difficult subset of GPQA and contains graduate-level, expert-written questions in biology, physics, and chemistry.
- **R-Bench** (Guo et al., 2025) evaluates graduate-level complex reasoning across multiple disciplines.
- **MBPP+** (Liu et al., 2023) evaluates code generation using an augmented version of MBPP with more rigorous test cases.
- **MathVista** (Lu et al., 2023) evaluates mathematical reasoning over visual and textual contexts.
- **MMMU-Math** (Yue et al., 2024) is the mathematical subset of MMMU and evaluates multimodal understanding and reasoning in expert-level settings.

Table 5: **Calibration comparison across five reasoning benchmarks.** Brier Score and ECE are reported. Asterisks mark multimodal benchmarks; **bold** marks the best average for each metric within each model family.

	GSM8K		MATH500		AIME25		MathVista*		MMMU-Math*		Average	
Method	**Brier ↓**	**ECE ↓**	**Brier ↓**	**ECE ↓**	**Brier ↓**	**ECE ↓**	**Brier ↓**	**ECE ↓**	**Brier ↓**	**ECE ↓**	**Brier ↓**	**ECE ↓**
Qwen3-8B & Qwen3-VL-8B*												
SC	0.081	0.052	0.083	0.051	0.146	0.171	0.108	0.116	0.195	0.132	**0.123**	0.104
CISC	0.108	0.109	0.109	0.104	0.718	0.724	0.255	0.258	0.189	0.187	0.276	0.276
CW-SC	0.080	0.058	0.086	0.077	0.141	0.159	0.108	0.105	0.202	0.134	**0.123**	0.106
AVG_{Trace} (Conservative)	0.082	0.056	0.082	0.068	0.161	0.129	0.106	0.117	0.189	0.137	0.124	**0.101**
AVG_{Trace} (Aggressive)	0.109	0.102	0.100	0.068	0.317	0.467	0.194	0.224	0.250	0.228	0.194	0.218
AVG_{Tail} (Conservative)	0.089	0.068	0.083	0.050	0.176	0.169	0.107	0.117	0.196	0.123	0.130	0.105
AVG_{Tail} (Aggressive)	0.120	0.109	0.102	0.084	0.242	0.450	0.179	0.193	0.241	0.234	0.177	0.214
VeTo (Conservative)	0.076	0.063	0.084	0.072	0.203	0.213	0.108	0.118	0.196	0.148	0.133	0.123
VeTo (Asymmetric)	0.075	0.062	0.083	0.074	0.203	0.213	0.110	0.118	0.185	0.133	0.131	0.120
VeTo (Aggressive)	0.098	0.098	0.085	0.087	0.160	0.140	0.135	0.148	0.227	0.206	0.141	0.136
Llama-3.1-8B & Llama-3.2-11B-Vision*												
SC	0.077	0.066	0.145	0.091	-	-	0.156	0.121	0.215	0.115	0.148	**0.098**
CISC	0.263	0.303	0.371	0.395	-	-	0.289	0.288	0.506	0.497	0.357	0.371
CW-SC	0.083	0.065	0.136	0.118	-	-	0.159	0.146	0.227	0.166	0.151	0.124
AVG_{Trace} (Conservative)	0.077	0.076	0.138	0.079	-	-	0.160	0.141	0.230	0.110	0.151	0.102
AVG_{Trace} (Aggressive)	0.089	0.078	0.206	0.220	-	-	0.322	0.404	0.350	0.400	0.242	0.276
AVG_{Tail} (Conservative)	0.078	0.062	0.148	0.084	-	-	0.158	0.131	0.224	0.113	0.152	**0.098**
AVG_{Tail} (Aggressive)	0.101	0.086	0.192	0.254	-	-	0.239	0.300	0.305	0.370	0.209	0.253
VeTo (Conservative)	0.086	0.098	0.147	0.090	-	-	0.157	0.114	0.249	0.158	0.160	0.115
VeTo (Asymmetric)	0.080	0.090	0.090	0.065	-	-	0.125	0.118	0.249	0.159	**0.136**	0.108
VeTo (Aggressive)	0.076	0.084	0.094	0.070	-	-	0.125	0.118	0.302	0.254	0.149	0.132
InternLM3-8B & InternVL3-8B*												
SC	0.052	0.039	0.110	0.071	0.128	0.175	0.187	0.138	0.243	0.171	0.144	**0.119**
CISC	0.064	0.064	0.121	0.126	0.748	0.777	0.244	0.222	0.358	0.333	0.307	0.304
CW-SC	0.052	0.047	0.114	0.086	0.119	0.179	0.184	0.150	0.250	0.153	0.144	0.123
AVG_{Trace} (Conservative)	0.051	0.031	0.106	0.062	0.125	0.205	0.193	0.154	0.242	0.176	**0.143**	0.126
AVG_{Trace} (Aggressive)	0.069	0.066	0.131	0.146	0.392	0.583	0.319	0.350	0.340	0.340	0.250	0.297
AVG_{Tail} (Conservative)	0.053	0.037	0.109	0.081	0.125	0.181	0.190	0.151	0.249	0.179	0.145	0.126
AVG_{Tail} (Aggressive)	0.064	0.052	0.133	0.122	0.314	0.530	0.277	0.280	0.346	0.360	0.227	0.269
VeTo (Conservative)	0.052	0.048	0.108	0.067	0.109	0.187	0.183	0.188	0.265	0.187	**0.143**	0.135
VeTo (Asymmetric)	0.052	0.048	0.108	0.067	0.109	0.187	0.174	0.160	0.274	0.204	**0.143**	0.133
VeTo (Aggressive)	0.048	0.048	0.086	0.086	0.197	0.260	0.174	0.163	0.296	0.248	0.160	0.161

C Confidence Refinement and Calibration Evaluation

For calibration evaluation, we refine the confidence score by combining voting consistency with verbalized confidence. Let a_{SC} be the answer predicted by SC before filtering. We denote by $R_{\mathrm{SC}}(\hat{a})$ the sampling ratio of the final answer $\hat{a}$ in the original unfiltered pool $\mathcal{C}$ and by $R_{\mathrm{filter}}(\hat{a})$ its consensus ratio in the filtered pool $\mathcal{C}_{\mathrm{final}}$. The refined confidence is

$$\mathrm{Conf}_{\mathrm{final}} = \begin{cases} \max(R_{\mathrm{SC}}(\hat{a}), R_{\mathrm{filter}}(\hat{a})), & \mathcal{C}_{\mathrm{high}} \neq \emptyset \text{ and } \hat{a} = a_{\mathrm{SC}}, \\ R_{\mathrm{SC}}(\hat{a}), & \text{otherwise.} \end{cases}$$

When a high-confidence pool exists and the filtered answer agrees with the original SC answer, the maximum preserves the larger of the pre- and post-filtering consensus ratios. In all other cases, the refined confidence equals the original sampling ratio.

C.1 Calibration Results

Empirical results in Table 5 show that the calibration effect varies across model families. The Asymmetric variant reduces the average Brier Score from 0.148 to 0.136 for the Llama family. For InternLM, its average Brier Score is 0.143 compared with 0.144 for SC, while ECE increases from 0.119 to 0.133. For the Qwen family, its average Brier Score increases from 0.123 to 0.131, and ECE

increases from 0.104 to 0.120. Overall, the Asymmetric variant improves the average Brier Score for Llama and slightly for InternLM, but does not consistently improve calibration across families.

D Additional Analyses and Comparisons

D.1 Correct Path Retention in the Low-Confidence Regime

The observations in Section 3.1 motivate examining which correct paths survive fallback filtering. We evaluate instances that actually enter the low-confidence branch for Llama-3.1-8B on MATH500 and Llama-3.2-11B-Vision on MathVista, using each model's own verbalized confidence. Correct path recall measures the fraction of correct paths retained after filtering. For each β, we compare the highest-jitter paths with the same number of lowest-jitter and randomly selected paths on each question.

Table 6: **Correct path recall under different fallback retention thresholds.** MATH500 and MathVista contain 155 and 201 fallback questions with 959 and 919 correct paths, respectively. Random reports the exact expected recall of equal-size random selections. Retained Paths denotes the realized proportion of candidates retained, which can differ from β because of discrete pool sizes and jitter ties ($N = 16$).

β	Retained Paths (%)	Random (%)	Low Jitter (%)	High Jitter (%)
Llama-3.1-8B / MATH500				
50	50.00	50.00	48.07	51.93
70	68.75	68.75	68.51	70.39
90	87.58	87.55	88.63	89.05
Llama-3.2-11B-Vision / MathVista				
50	79.10	85.80	84.33	87.27
70	91.29	95.48	94.99	96.19
90	97.76	98.76	98.04	99.35

At all three thresholds $\beta \in \{50, 70, 90\}$, high-jitter filtering exceeds both equal-size random selection and low-jitter filtering in correct path recall in both settings. These results support high-jitter retention as a useful fallback heuristic in these settings.

Table 7 further examines whether high-jitter retention benefits the final vote on fallback instances at $\beta = 90$. Compared with no filtering, size-matched random filtering, and low-jitter filtering, High Jitter achieves the highest accuracy in both settings, improving over no filtering from 57.42% to 58.06% on MATH500 and from 34.33% to 34.83% on MathVista. These results complement the recall analysis by showing that high-jitter retention can also benefit downstream voting accuracy.

Table 7: **Final voting accuracy on fallback instances.** Accuracy (%) is reported for the instances in Table 6 at $\beta = 90$.

Model / Benchmark	No Filtering	Random	Low Jitter	High Jitter
Llama-3.1-8B / MATH500	57.42	56.33	56.77	58.06
Llama-3.2-11B-Vision / MathVista	34.33	34.27	33.83	34.83

D.2 Token Selection within the Framework

To examine the choice of token selection rule within our framework, we construct two matched controls by replacing jitter-based selection with average token confidence or random pruning. Both controls retain our verbalized confidence pooling procedure and match the number of paths retained by VeTo on each question. When no high-confidence path exists, both controls apply their selection rules to the full sampled pool. VC uses the resulting pool without further pruning; VC + Average

Token Confidence ranks candidates by geometric mean token probability, while VC + Random Pruning samples them uniformly. Random pruning reports the mean over 1,000 repetitions. The results show that VeTo (Conservative) achieves the highest average accuracy among these variants. We assess uncertainty in the differences between these matched selection rules using paired analyses.

Table 8: **Comparison of token selection variants within our framework.** Accuracy (%) is reported. VC denotes verbalized confidence pooling without further pruning. Average is computed across all eight benchmarks. Asterisks mark multimodal benchmarks; **bold** marks the best average.

Method	GSM8K	MATH500	AIME25	GPQA-D.	R-Bench	MBPP+	MathVista*	MMMU-Math*	Average
SC	87.6	86.8	26.7	45.5	15.6	66.0	69.2	61.2	57.3
VC	88.0	88.4	26.7	47.5	16.8	67.2	70.0	61.2	58.2
VC + Average Token Confidence	88.0	89.2	30.0	47.0	17.2	66.8	69.2	61.6	58.6
VC + Random Pruning	88.1	88.7	26.6	47.4	17.5	67.2	70.0	61.2	58.3
VeTo (Conservative)	88.0	90.0	33.3	50.0	19.2	68.0	69.6	63.2	**60.2**

To quantify uncertainty in these gains, we conduct paired analyses over the 1,728 questions evaluated with Qwen. We perform 100,000 paired bootstrap resamples within each benchmark and average the accuracy differences equally across the eight benchmarks, matching Table 8. VeTo improves over average token confidence filtering by 1.545% (95% bootstrap percentile confidence interval: $[0.692, 2.653]\%$) and over expected random-pruning accuracy by 1.820% ($[0.940, 2.877]\%$). For random pruning, the expected accuracy is computed from each question's mean correctness over 1,000 random selections. A two-sided paired permutation test with 200,000 label swaps and the same benchmark weighting yields $p = 4.5 \times 10^{-5}$ for the comparison with average token confidence.

Moreover, with the questions and candidate pools fixed, VeTo exceeds all 1,000 random-pruning runs. This comparison isolates pruning randomness from variation due to newly generated paths.

Qwen evaluator for Llama paths. We repeat these controls on the seven available benchmarks in Table 2, using the same 16 paths per question, Qwen confidence scores, and Llama's two window sizes without further tuning. VeTo (Conservative) reaches 55.3% average accuracy, compared with 54.8% for VC, 54.4% for average token confidence filtering, and 54.5% for random pruning. Applying the same paired analysis to these 1,698 questions gives gains of 0.502% over VC (95% CI $[-0.271, 1.275]\%$, $p = 0.219$), 0.902% over average token confidence filtering ($[0.027, 1.789]\%$, $p = 0.0496$), and 0.865% over random pruning ($[0.233, 1.506]\%$, $p = 0.0077$). Against average token confidence, VeTo improves five benchmarks and declines on MBPP+ and MathVista.

These comparisons separate the shared confidence-guided design from the choice of token selection rule: jitter achieves the highest observed average in both settings, with positive paired intervals against average token confidence under the tested configurations.

Table 9: **Binary and numeric verbalized confidence elicitation.** Accuracy (%) is reported for Qwen3-8B under the Conservative variant ($\alpha = 10$, $\beta = 90$); **bold** marks the best average.

Method	τ	GSM8K	MATH500	AIME25	GPQA-Diamond	Average
VeTo (binary VC)	100	88.0	90.0	33.3	50.0	65.3
VeTo (numeric VC)	80	88.0	89.2	36.7	43.4	64.3
	90	88.0	89.6	36.7	47.5	**65.4**
	95	88.0	89.6	36.7	47.5	**65.4**
	99	89.2	87.6	30.0	46.0	63.2
	100	89.2	87.6	30.0	46.0	63.2

D.3 Verbalized Confidence Elicitation

The main VeTo prompts (Appendix Section F) elicit binary verbalized confidence in $\{0, 100\}$, making $\tau = 100$ the natural boundary without an additional threshold. To test sensitivity beyond binary scores, we elicit numeric confidence and vary τ on four Qwen3-8B text benchmarks while fixing $\alpha = 10$, $\beta = 90$, and the main window settings. Table 9 shows no uniformly best cutoff;

$\tau \in \{90, 95\}$ gives the highest average, while $\tau = 99$ and 100 produce identical results. For numeric VC at $\tau = 100$, the fractions of questions with a nonempty high-confidence pool are 97.2%, 94.4%, 60.0%, and 37.4%, and the fractions of individual paths assigned a score of 100 are 84.4%, 77.9%, 21.3%, and 12.0% for GSM8K, MATH500, AIME25, and GPQA-Diamond, respectively. Numeric VC therefore remains comparable to the binary formulation while exhibiting substantial cross-benchmark variation.

D.4 Local Group Confidence Baselines

Table 10 compares VeTo with two local group confidence baselines at group sizes of 1,024 and 2,048 tokens. VeTo improves over the strongest configuration by 2.46%, 2.06%, and 0.87% for the Qwen, Llama, and InternLM families, respectively. The closely clustered baseline results indicate that the gains are not tied to a particular group size or selection rule.

Table 10: **Local group confidence baselines versus VeTo (Conservative).** Accuracy (%) is reported. Parentheses give the local baseline group size in tokens; VeTo retains its main window settings. Asterisks mark multimodal benchmarks; dashes denote unavailable results; **bold** marks the best average within each model family.

Method	GSM8K	MATH500	AIME25	GPQA-D.	R-Bench	MBPP+	MathVista*	MMMU-Math*	Average
Qwen3-8B & Qwen3-VL-8B									
Bottom 10% Group (1,024)	87.6	87.2	30.0	44.4	15.6	65.6	68.8	62.0	57.66
Lowest Group (1,024)	87.6	87.6	30.0	44.4	15.6	65.6	68.8	62.0	57.71
Bottom 10% Group (2,048)	87.6	87.2	30.0	45.5	15.2	65.6	68.8	61.6	57.68
Lowest Group (2,048)	87.6	87.2	30.0	45.5	14.8	65.6	68.8	61.6	57.63
VeTo (Conservative)	88.0	90.0	33.3	50.0	19.2	68.0	69.6	63.2	**60.17**
Llama-3.1-8B & Llama-3.2-11B-Vision									
Bottom 10% Group (1,024)	91.2	68.4	—	28.3	44.4	56.4	39.2	38.4	52.33
Lowest Group (1,024)	91.2	68.4	—	27.8	45.2	56.4	39.2	38.8	52.43
Bottom 10% Group (2,048)	91.2	67.6	—	28.3	43.6	56.4	39.2	38.4	52.10
Lowest Group (2,048)	91.2	68.0	—	28.3	43.6	56.4	39.2	38.4	52.15
VeTo (Conservative)	91.2	69.6	—	33.8	50.0	56.8	38.8	41.2	**54.49**
InternLM3-8B & InternVL3-8B									
Bottom 10% Group (1,024)	93.2	84.4	16.7	39.9	55.6	60.0	56.8	46.4	56.62
Lowest Group (1,024)	93.2	84.4	16.7	39.9	55.2	60.0	56.8	46.4	56.57
Bottom 10% Group (2,048)	93.2	84.4	16.7	40.4	55.6	60.0	56.8	46.4	56.68
Lowest Group (2,048)	93.2	84.4	16.7	40.4	55.6	60.0	56.8	46.4	56.68
VeTo (Conservative)	94.4	84.4	16.7	42.9	59.2	59.6	53.6	49.6	**57.55**

E Hyperparameter Ablations

E.1 Alpha and Beta Ablations

Tables 11 and 12 report α and β ablations over three pooled benchmark groups. MATH includes GSM8K, MATH500, and AIME25, with AIME25 omitted because its results are unavailable for Llama; MULTIDISC includes GPQA-Diamond and R-Bench; and VISION includes MathVista and MMMU-Math. Group-level accuracy is computed over all instances pooled within each group. MBPP+ is excluded from these ablations.

The results show that moderate filtering of the high-confidence pool improves over $\alpha = 0$ in most settings, while aggressive filtering can reduce accuracy. The β ablation shows that selective fallback retention can outperform no fallback filtering ($\beta = 100$), but the benefit varies across model families and task categories.

Table 11: **Ablation of α across three benchmark groups.** Accuracy (%) is reported with $\beta = 90$. Δ denotes the accuracy gain of VeTo over SC; **bold** marks the best Δ within each model family and benchmark group.

Group	α	Qwen3/Qwen3-VL			Llama-3.1/Llama-3.2-Vision			InternLM3/InternVL3		
		SC	VeTo	Δ	SC	VeTo	Δ	SC	VeTo	Δ
MATH	0	83.77	84.72	+0.94	79.80	80.20	+0.40	84.72	85.09	+0.38
	10		85.85	**+2.08**		80.40	**+0.60**		85.28	**+0.57**
	30		85.47	+1.70		79.80	+0.00		84.72	+0.00
	50		85.28	+1.51		79.60	-0.20		84.53	-0.19
	70		83.96	+0.19		79.20	-0.60		83.58	-1.13
	90		82.83	-0.94		79.60	-0.20		81.89	-2.83
MULTIDISC	0	28.79	30.80	+2.01	38.39	40.62	+2.23	48.66	50.00	+1.34
	10		32.81	**+4.02**		42.86	+4.46		52.01	+3.35
	30		32.14	+3.35		42.63	+4.24		52.68	**+4.02**
	50		32.14	+3.35		42.63	+4.24		50.89	+2.23
	70		32.14	+3.35		43.08	**+4.69**		49.11	+0.45
	90		32.59	+3.79		40.85	+2.46		47.32	-1.34
VISION	0	65.20	65.60	+0.40	38.20	39.00	+0.80	51.80	51.80	**+0.00**
	10		66.40	**+1.20**		40.00	**+1.80**		51.60	-0.20
	30		66.00	+0.80		38.80	+0.60		50.80	-1.00
	50		66.40	**+1.20**		39.40	+1.20		50.60	-1.20
	70		65.60	+0.40		38.60	+0.40		49.80	-2.00
	90		64.60	-0.60		37.40	-0.80		48.80	-3.00

Table 12: **Ablation of β across three benchmark groups.** Accuracy (%) is reported with $\alpha = 10$. Δ denotes the accuracy gain of VeTo over SC; **bold** marks the best Δ within each model family and benchmark group.

Group	β	Qwen3/Qwen3-VL			Llama-3.1/Llama-3.2-Vision			InternLM3/InternVL3		
		SC	VeTo	Δ	SC	VeTo	Δ	SC	VeTo	Δ
MATH	10	83.77	85.85	**+2.08**	79.80	70.40	-9.40	84.72	85.28	**+0.57**
	30		85.85	**+2.08**		77.60	-2.20		85.28	**+0.57**
	50		85.85	**+2.08**		79.60	-0.20		85.28	**+0.57**
	70		85.85	**+2.08**		79.40	-0.40		85.28	**+0.57**
	90		85.85	**+2.08**		80.40	**+0.60**		85.28	**+0.57**
	100		85.85	**+2.08**		79.80	+0.00		85.28	**+0.57**
MULTIDISC	10	28.79	31.25	+2.46	38.39	44.20	+5.80	48.66	51.12	+2.46
	30		32.37	+3.57		44.20	+5.80		52.01	**+3.35**
	50		31.92	+3.12		43.97	+5.58		51.79	+3.12
	70		33.26	**+4.46**		44.87	**+6.47**		51.79	+3.12
	90		32.81	+4.02		42.86	+4.46		52.01	**+3.35**
	100		32.37	+3.57		42.41	+4.02		52.01	**+3.35**
VISION	10	65.20	66.40	+1.20	38.20	37.40	-0.80	51.80	53.20	**+1.40**
	30		67.20	**+2.00**		39.60	+1.40		52.00	+0.20
	50		67.00	+1.80		39.60	+1.40		51.80	+0.00
	70		66.60	+1.40		40.00	**+1.80**		51.20	-0.60
	90		66.40	+1.20		40.00	**+1.80**		51.60	-0.20
	100		66.40	+1.20		39.80	+1.60		51.40	-0.40

E.2 Window Size Sensitivity

For the main experiments, each model family uses one window size for text benchmarks (including MBPP+) and one for vision benchmarks. The text/vision settings are $(60, 5)$ for Qwen, $(20, 95)$ for Llama, and $(105, 75)$ for InternLM. Each pair is shared across all benchmarks of the corresponding modality and all three VeTo variants. Scaling experiments retain the same family-level settings across budgets and model sizes; Qwen-evaluated Llama paths retain the Llama window sizes. Figure 8 reports sensitivity to alternative window sizes.

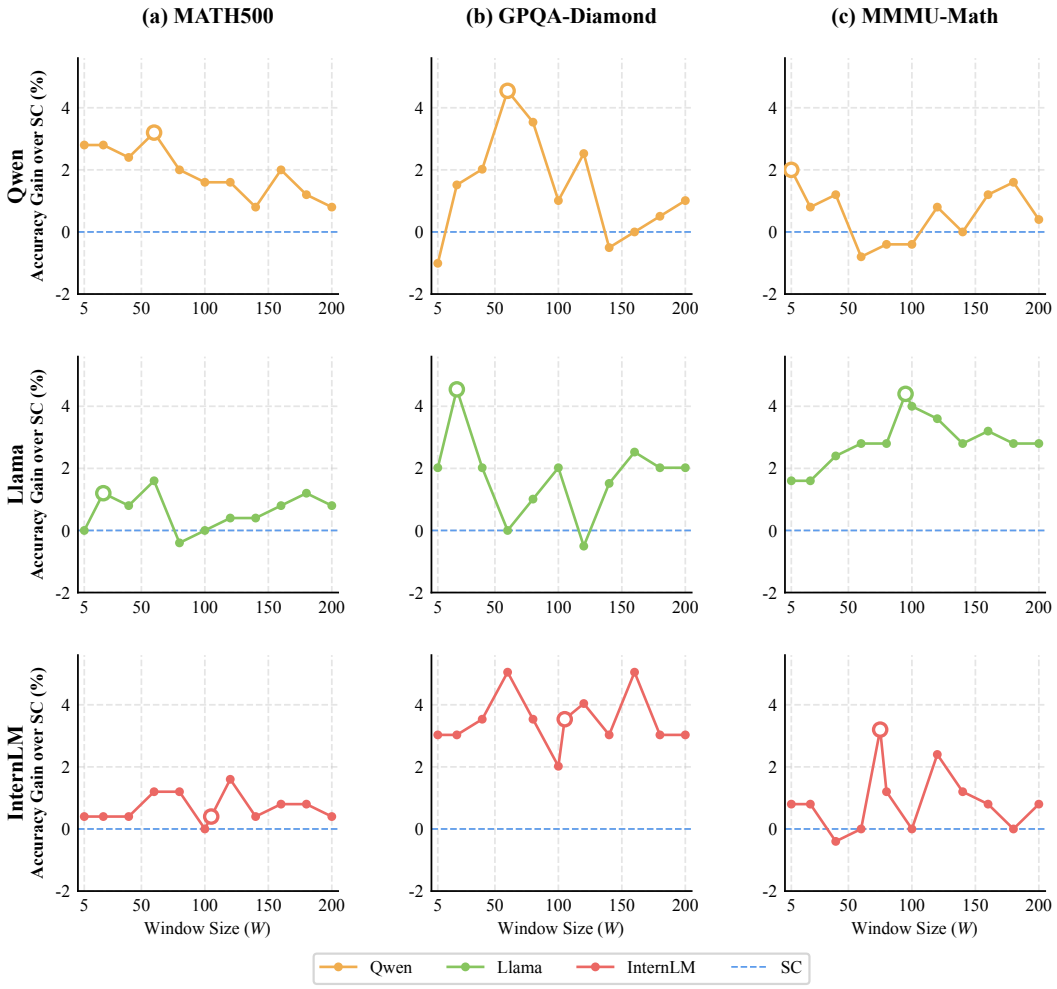


Figure 8: **Window size sensitivity.** Accuracy gains over SC (%) are reported for the Conservative variant ($\alpha = 10$, $\beta = 90$). Rows correspond to Qwen, Llama, and InternLM; columns correspond to MATH500, GPQA-Diamond, and MMMU-Math. Dashed lines mark SC, and outlined circles indicate the main-experiment window settings.

F BINARY VERBALIZED CONFIDENCE PROMPTS

We use the following prompts for mathematical and non-mathematical reasoning, respectively. Both specify binary confidence scores of 0 or 100.

Binary VC prompt for mathematical reasoning

Analyze the current thought process against the problem description. There is NO room for ambiguity in Grade School Math.

CRITERIA:
- **100%**: The step clearly follows from the previous sentence and the numbers align perfectly with the problem.
- **0%**:

1. Any number appears without a clear source (Magic Numbers).
2. The operation doesn't make sense in the real-world context.
3. The calculation is wrong.

Note: Be harsh. If the logic feels 'jumpy' or 'weird', give 0%.

Your Turn:
Output strictly in this format: **Confidence: [Score]%**

Binary VC prompt for non-mathematical reasoning

Analyze the current thought process against the problem constraints. Logic must be sound and consistent.

CRITERIA:
- **100%**: The reasoning step represents a valid logical deduction that advances towards the solution.
- **0%**:

1. Introduces numbers or assumptions without basis (Hallucinations).
2. The logic contradicts the problem statement or previous steps.
3. The reasoning method is fundamentally flawed or leads to a dead end.

Note: Be harsh. If the thought process feels 'drifting' or 'inconsistent', give 0%.

Your Turn:
Output strictly in this format: **Confidence: [Score]%**