BEYOND FINAL CONFIDENCE: LEVERAGING CONFIDENCE DYNAMICS FOR TEST-TIME REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time scaling improves language model reasoning by sampling and aggregating multiple reasoning trajectories. Recent methods further leverage model confidence to prioritize reliable trajectories. However, final confidence alone can be misleading, as failed trajectories may remain highly confident while repeatedly following an incorrect reasoning direction with little effective self-correction. We refer to this failure mode as *high-confidence reasoning stagnation*. To distinguish trajectories within the high-confidence regime, we measure the change in token confidence from the penultimate to the final layer and aggregate it over the tail of each trajectory. Among high-confidence samples, stronger inter-layer confidence dynamics are associated with higher accuracy, shorter reasoning trajectories, and less observed stagnation, providing a complementary signal beyond final confidence. Based on this observation, we introduce a training-free aggregation method that jointly filters trajectories using tail confidence and inter-layer confidence dynamics, and then weights the retained trajectories by their confidence dynamics. Across Qwen3-1.7B, Qwen3-8B, and Gemma4-12B, our method consistently improves overall accuracy over standard majority voting and confidence-based aggregation on the evaluated reasoning benchmarks. Further analyses show that these gains remain robust under increasing sampling budgets, while confidence dynamics are most informative when measured across layers close to the final readout.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated strong reasoning capabilities through test-time scaling, where multiple reasoning trajectories are sampled and aggregated via strategies such as self-consistency and majority voting (Wang et al., 2022; Brown et al., 2024). Recent methods further leverage model-internal signals during generation, such as output probability and entropy (Farquhar et al., 2024; Kang et al., 2025; Fu et al., 2026b), to estimate trajectory quality and prioritize more reliable reasoning paths. By filtering or downweighting low-confidence trajectories, these approaches can improve both reasoning accuracy and test-time efficiency.

However, confidence remains only a coarse indicator of reasoning quality. Confidence-based selection can saturate in the high-confidence range, where increasingly high confidence no longer implies increasingly reliable reasoning and provides limited discrimination among trajectories (Wan et al., 2025; Hiremath & Hiremath, 2026). As illustrated in Figure 1, successful and failed trajectories can exhibit similarly high confidence, particularly toward the end of reasoning, while showing fundamentally different reasoning behaviors. The successful trajectory continues to make meaningful intermediate progress before reaching a correct answer. In contrast, the failed trajectory commits to an incorrect hypothesis and subsequently spends substantial computation reasoning around or reinforcing the same direction, with little effective correction, while remaining highly confident. We refer to this failure mode as *high-confidence reasoning stagnation*. Consequently, static confidence alone cannot distinguish productive reasoning from persistent commitment to an incorrect solution.

This observation motivates us to look beyond how confident a trajectory is and examine how confidence evolves through the model's late-stage computation. Specifically, we investigate the confidence change from the penultimate to the final layer at each generation step as inter-layer confidence dynamics, capturing how much the model's final computation updates its prediction confidence. Our investigation in Sec. 3.3 reveals a clear distinction even among similarly high-confidence trajectories: stronger confidence dynamics are associated with higher correctness, shorter reasoning trajectories,

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.

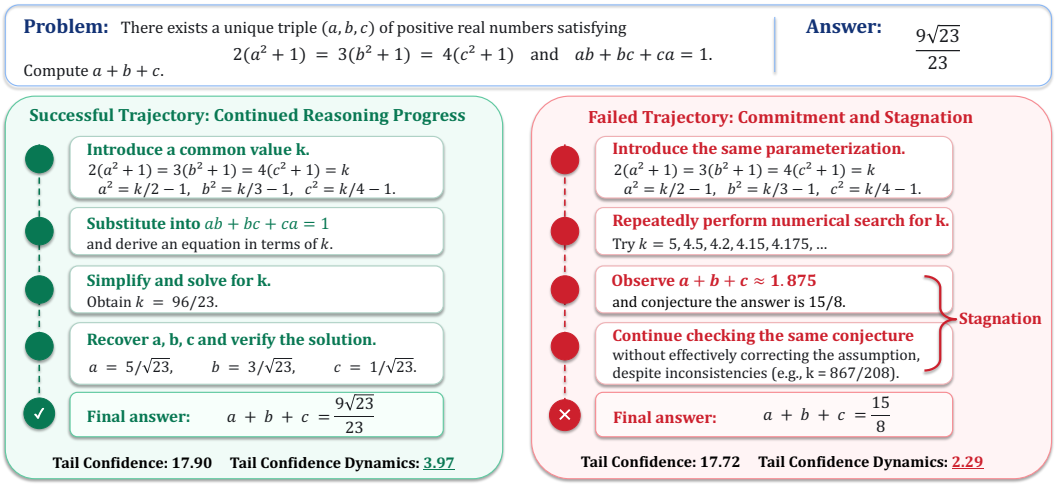


Figure 1: **Illustration of high-confidence reasoning stagnation.** Two Qwen3-8B trajectories for the same HMMT2025 (Dekoninck et al., 2026) problem exhibit similar tail confidence but distinct reasoning behaviors. The successful trajectory makes continued progress toward the correct answer, while the failed trajectory commits to an incorrect conjecture and stagnates with little effective correction. Their distinct tail confidence dynamics complement static confidence, with detailed token-level confidence and dynamics provided in Appendix A.

and substantially less reasoning stagnation, whereas weaker dynamics frequently accompany prolonged but ineffective computation. This suggests that inter-layer confidence dynamics provides a complementary signal to static confidence, of whether a confident trajectory continues to undergo meaningful computational updating.

Based on these observations, we introduce a simple training-free, confidence-dynamics-aware method for test-time aggregation. Rather than collapsing confidence and confidence dynamics into a single score, we use them as complementary selection criteria: we retain trajectories that rank highly under both confidence and inter-layer confidence dynamics, and then aggregate their answers using confidence dynamics as voting weights. We focus both signals on the tail portion of each trajectory, where reasoning degradation and stagnation are more likely to emerge in long reasoning chains. This design filters out both uncertain trajectories and high-confidence trajectories exhibiting weak late-stage updating, without requiring external reward models or additional training.

We evaluate our method on Qwen3-1.7B, Qwen3-8B (Yang et al., 2025), and Gemma4-12B (Team et al., 2026) across various reasoning benchmarks, including HMMT (2025 February & November) (Dekoninck et al., 2026), GPQA-Diamond (Rein et al., 2023), OlympiadBench-Math (He et al., 2024), Countdown (Xing et al., 2026), and SuperGPQA (Du et al., 2025). Our approach achieves higher overall accuracy than standard majority voting and confidence-based aggregation across all evaluated models and sampling budgets. These results highlight that reliable test-time reasoning depends not only on how confident a model ultimately becomes, but also on whether that confidence is accompanied by continued internal computational updating.

2 Preliminaries

We first formalize the confidence measures and confidence-based test-time scaling strategies used in this work. We compute confidence from the model's token-level output distributions and aggregate it over reasoning trajectories, and introduce confidence-based weighting and filtering for test-time scaling. We then examine the limitations of confidence as an indicator of reasoning quality, highlighting a key failure mode that motivates our approach.

2.1 Confidence Measurement

Token Confidence. Given an autoregressive language model generating a reasoning trajectory $\tau = (y_1, \ldots, y_T)$, let $p_t^{(L)} \in \mathbb{R}^{|\mathcal{V}|}$ denote the next-token probability distribution at decoding step t, obtained from the final layer L and $\mathcal{V}$ is the vocabulary. Following DeepConf (Fu et al., 2026b),

token confidence is defined as the negative average log-probability of the top-k candidate tokens:

$$C_t^{(L)} = -\frac{1}{k}\sum_{j=1}^{k} \log p_t^{(L)}(j). \tag{1}$$

where $p_t^{(L)}(j)$ denotes the probability of the j-th ranked token. A higher confidence score corresponds to more concentrated output distributions and greater model certainty. Unless otherwise specified, we use $k = 20$ following DeepConf (Fu et al., 2026b).

Trajectory-Level Confidence Aggregation. To assess a sequence's overall uncertainty, token-level confidence can be aggregated globally over the full trajectory with T tokens or locally over a tail window $\mathcal{T}_{\text{tail}}(\tau)$:

$$C_{\text{avg}}(\tau) = \frac{1}{T}\sum_{t=1}^{T} C_t^{(L)}, \qquad C_{\text{tail}}(\tau) = \frac{1}{|\mathcal{T}_{\text{tail}}(\tau)|}\sum_{t\in\mathcal{T}_{\text{tail}}(\tau)} C_t^{(L)}. \tag{2}$$

The tail window can be defined either as a fixed number or proportion of tokens. While $C\text{avg}$ summarizes confidence throughout the trajectory, C_{tail} focuses on late-stage behavior and is the best-performing aggregation in DeepConf (Fu et al., 2026b), as long reasoning chains are most prone to degradation or failure. We use $C(\tau)$ to denote trajectory-level confidence in general.

2.2 Confidence-Based Test-Time Scaling

Given an input x, test-time scaling samples N reasoning trajectories $\mathcal{S}(x) = \{(\tau_i, a_i)\}_{i=1}^{N}$, where τ_i is the i-th trajectory and a_i is its extracted final answer. Standard majority voting treats all samples equally: $\hat{a} = \arg\max_a \sum_{i=1}^{N} \mathbb{I}[a_i = a]$. Confidence-based approaches instead leverage trajectory confidence to weigh or filter candidates.

Confidence-Weighted Majority Voting. Each trajectory's vote is weighted by its confidence score $w_i = C(\tau_i)$:

$$\hat{a} = \arg\max_a \sum_{i=1}^{N} w_i \mathbb{I}[a_i = a]. \tag{3}$$

This strategy gives greater influence to answers supported by higher-confidence trajectories.

Confidence-Filtered Majority Voting. An alternative is to use confidence to select trajectories before aggregation. Let $\mathcal{SC} = \text{Top}\, m(\mathcal{S}(x); C)$ denote the m highest-confidence trajectories. Majority voting is then performed only over this subset:

$$\hat{a} = \arg\max_a \sum_{i:(\tau_i,a_i)\in\mathcal{S}_C} \mathbb{I}[a_i = a]. \tag{4}$$

Both strategies assume that higher confidence reflects greater reasoning quality for test-time scaling.

2.3 When Does Static Confidence Fail?

Despite its empirical success, confidence provides only a static measure of predictive certainty and may not fully reflect reasoning reliability. We therefore examine where confidence succeeds and where it falls short as an indicator of reasoning quality.

We first compare trajectory-level confidence distributions between correct and incorrect trajectories. As shown in Figure 2**(a,b)**, correct trajectories are generally more confident than incorrect ones under both global and tail aggregation. However, their distributions substantially overlap, with many incorrect trajectories also attaining high confidence. More importantly, this distinction rapidly saturates in the high-confidence regimes. Figure 2**(c)** shows that restricting trajectories to the top 30% by confidence initially improves accuracy, whereas further tightening the threshold yields little additional benefit for tail confidence and lower accuracy for global confidence. These results suggest that confidence is effective for filtering less reliable trajectories at a coarse level, but increasingly high confidence does not imply increasingly reliable reasoning. This motivates a complementary signal that captures how confidence changes through the model's internal computation.

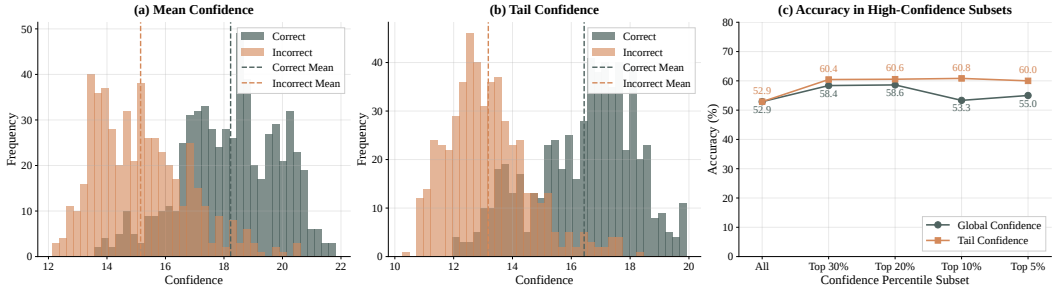


Figure 2: **Static confidence is informative at a coarse level, but saturates in high-confidence regimes.** (a,b) Distributions of mean and tail confidence (last 2048 tokens) for correct vs. incorrect trajectories generated by Qwen3-8B on HMMT25 Feb. & Nov. (60 problems, 20 sampled trajectories per problem). While correct trajectories exhibit higher confidence on average, substantial overlap remains. (c) Accuracy when evaluating increasingly confident trajectory subsets. Filtering out low-confidence samples initially boosts performance, but tightening the threshold beyond the top 30% yields no further gains and even reduces accuracy.

3 METHODOLOGY

We complement static final-layer confidence with *inter-layer confidence dynamics*, examining whether the model's prediction confidence continues to change during late-stage computation or remains largely unchanged. We define a token-level measure of inter-layer confidence change, aggregate it over each trajectory, and combine it with confidence for joint filtering and weighted aggregation.

3.1 MEASURING INTER-LAYER CONFIDENCE DYNAMICS

Token-Level Inter-Layer Confidence Change. At generation step t, standard confidence uses only the probability distribution produced by the final-layer representation $h_t^{(L)}$. To capture how the confidence evolves during the model's late-stage computation, we additionally read out the penultimate-layer representation $h_t^{(L-1)}$ using the same language modeling head W_{LM}:

$$p_t^{(\ell)} = \mathrm{softmax}\left(W_{\mathrm{LM}} h_t^{(\ell)}\right), \qquad \ell \in \{L-1, L\}. \tag{5}$$

Both layers apply the final normalization before the LM head, omitted from Eq. 5. We derive layer-wise confidences $C_t^{(L-1)}$ and $C_t^{(L)}$ from the two distributions using Eq. 1 and define the inter-layer confidence change as:

$$\Delta C_t = C_t^{(L)} - C_t^{(L-1)}. \tag{6}$$

We use the penultimate layer as the reference because it is closest to the final readout, while earlier-layer representations are less aligned with the output space and introduce greater readout mismatch when projected through the LM head. The resulting inter-layer confidence change can be viewed as a first-order finite difference of confidence across layers, capturing the confidence change introduced by the final layer. We further analyze confidence changes using earlier reference layers in Section 4.6.

Trajectory-Level Confidence Dynamics Aggregation. For a trajectory τ of length T, the token-wise measurements form a confidence-dynamics sequence $\Delta\mathbf{C}(\tau) = (\Delta C_1, \ldots, \Delta C_T)$ globally or over a tail window $\mathcal{T}_{\mathrm{tail}}(\tau)$:

$$\Delta C_{\mathrm{avg}}(\tau) = \frac{1}{T}\sum_{t=1}^{T} \Delta C_t, \qquad \Delta C_{\mathrm{tail}}(\tau) = \frac{1}{|\mathcal{T}_{\mathrm{tail}}(\tau)|}\sum_{t \in \mathcal{T}_{\mathrm{tail}}(\tau)} \Delta C_t. \tag{7}$$

Evaluated over identical windows, $C(\tau)$ measures how confident the model is, whereas $\Delta C(\tau)$ reflects how strongly that confidence continues to change through late-stage computation.

3.2 COMBINING CONFIDENCE AND CONFIDENCE DYNAMICS FOR MAJORITY VOTING

Joint Confidence-Dynamics Filtering. Given a pool of N sampled trajectories $\mathcal{S}(x) = \{(\tau_i, a_i)\}_{i=1}^{N}$, we independently rank them according to their trajectory-level confidence $C(\tau_i)$ and inter-layer confidence dynamics $\Delta C(\tau_i)$. For a retention ratio ρ, we construct

$$\mathcal{S}_C^{\rho} = \mathrm{Top}_{\rho}\big(\mathcal{S}(x); C(\tau)\big), \qquad \mathcal{S}_{\Delta}^{\rho} = \mathrm{Top}_{\rho}\big(\mathcal{S}(x); \Delta C(\tau)\big), \tag{8}$$

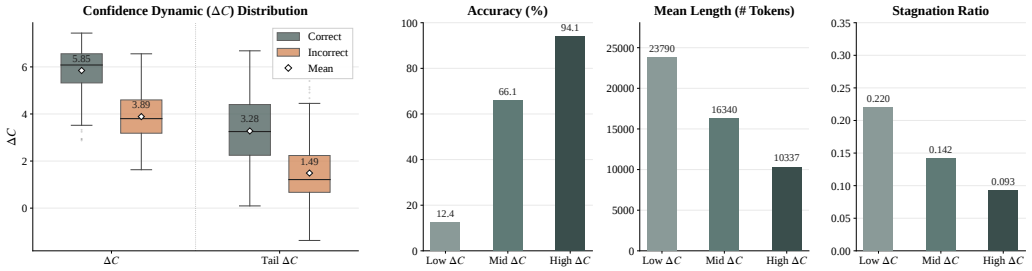


Figure 3: **Confidence dynamics within the top-50% high-confidence (global) trajectories** of Qwen3-8B on HMMT25 Feb. & Nov. **Left:** distributions of global and tail confidence dynamics (ΔC) for correct and incorrect trajectories. **Right:** trajectories are divided into three equal-sized groups according to their global ΔC (Low, Mid, and High); we report answer accuracy, mean sequence length, and the proportion of reasoning segments labeled as stagnation. Higher confidence dynamics are associated with higher accuracy, shorter trajectories, and less reasoning stagnation.

where Top_ρ retains the top ρ proportion of trajectories ranked by the corresponding signal. We then take the intersection of the two subsets:

$$\mathcal{S}^{\rho}_{C,\Delta} = \mathcal{S}^{\rho}_{C} \cap \mathcal{S}^{\rho}_{\Delta}. \tag{9}$$

This intersection requires a retained trajectory to exhibit both high predictive confidence and substantial inter-layer confidence dynamics. In this way, confidence dynamics act as a complementary criterion that can exclude highly confident trajectories exhibiting weak late-stage updating, while confidence prevents trajectories with strong dynamics but low overall certainty from being selected.

Confidence-Dynamics-Weighted Majority Voting. After joint filtering, we further use confidence dynamics to differentiate the contributions of the retained trajectories. The final prediction is obtained by dynamics-weighted voting:

$$\hat{a}_{\mathrm{CD}} = \arg\max_{a} V_{\mathrm{CD}}(a) = \arg\max_{a} \sum_{i:(\tau_i,a_i)\in\mathcal{S}^{\rho}_{C,\Delta}} \Delta C(\tau_i)\, \mathbb{I}[a_i = a]. \tag{10}$$

Thus, confidence and confidence dynamics play distinct roles in our framework: both signals determine which trajectories are retained, while confidence dynamics further determine their relative contributions during answer aggregation. The entire procedure is training-free and requires no external verifier, reward model, or additional supervision.

3.3 Analyzing High-Confidence Dynamics

We examine the correlation between inter-layer confidence dynamics and reasoning correctness within the top-50% of trajectories ranked by global confidence on Qwen3-8B and HMMT25 Feb. & Nov. As shown on the left of Figure 3, correct trajectories consistently exhibit higher confidence dynamics than incorrect ones under both global and tail aggregation. The separation becomes particularly pronounced toward the end of reasoning where the mean tail ΔC is 3.28 for correct trajectories, compared with only 1.49 for incorrect ones. This suggests that confidence dynamics distinguish the two groups even within this high-confidence subset.

To further characterize this relationship, we divide the high-confidence subset into three equal-sized groups, i.e., Low, Mid, and High, according to global ΔC. As shown on the right of Figure 3, as ΔC increases, accuracy rises substantially from 12.4% to 94.1%, while the average sequence length decreases from 23.8K to 10.3K tokens. To examine whether these differences are associated with reasoning behavior, we further segment each reasoning trajectory and use Gemma4-12B to classify each segment into one of four functional categories: *progress*, *verification*, *revision or strategy switch*, and *stagnation*. The proportion of segments classified as stagnation decreases monotonically from 22.0% in the low-ΔC group to 9.3% in the high-ΔC group. Together, these results show that weaker confidence dynamics are associated not only with lower correctness, but also with longer reasoning trajectories and substantially more stagnant computation. Details of trajectory segmentation and segment-level functional classification are provided in Appendix C.

Dataset	Standard	Conf. W.	Conf. F. 30%	Conf. F. 50%	Tail Conf. W.	Tail Conf. F. 30%	Tail Conf. F. 50%	Ours
Qwen3-1.7B								
HMMT	31.67	31.67	31.67	35.00	31.67	33.33	33.33	**35.00**
GPQA-D	35.35	34.85	**38.38**	37.88	34.85	36.87	36.87	**38.38**
Olympiad	70.96	71.41	70.37	72.00	71.56	70.22	72.15	**72.59**
Countdown	74.00	75.00	81.67	79.67	75.00	82.67	81.00	**83.00**
Overall	53.00	53.23	55.52	56.14	53.27	55.77	55.84	**57.24**
Δ	-	+0.23	+2.52	+3.14	+0.27	+2.77	+2.84	+4.24
Qwen3-8B								
HMMT	63.33	**66.67**	60.00	65.00	**66.67**	60.00	61.67	**66.67**
GPQA-D	60.61	61.62	61.11	61.11	**62.12**	61.62	61.11	**62.12**
Olympiad	81.48	81.33	81.19	81.04	81.04	81.48	81.33	**81.78**
Countdown	77.67	79.00	88.00	85.67	79.67	88.33	85.67	**89.67**
Overall	70.77	72.16	72.58	73.21	72.38	72.86	72.45	**75.06**
Δ	-	+1.39	+1.81	+2.44	+1.61	+2.09	+1.68	+4.29
Gemma4-12B								
HMMT	86.67	85.00	83.33	83.33	85.00	85.00	85.00	**88.33**
GPQA-D	77.78	77.27	75.25	77.27	**78.28**	76.77	77.27	**78.28**
Olympiad	85.48	**85.78**	85.48	**85.78**	**85.78**	85.48	**85.78**	85.48
SuperGPQA	56.00	55.50	**58.50**	57.50	55.50	57.50	56.50	57.50
Overall	76.48	75.89	75.64	75.97	76.14	76.19	76.14	**77.40**
Δ	-	-0.59	-0.84	-0.51	-0.34	-0.29	-0.34	+0.92

Table 1: Main results of different majority voting strategies. We report accuracy (%) on HMMT, GPQA-Diamond (GPQA-D), Olympiad, and Countdown/SuperGPQA. **Overall** denotes the macro-average across the four benchmarks. The best result for each model and dataset is shown in bold. **Δ** is the improvement of different strategies over the standard majority voting over all valid trajectories with uniform weights.

4 Experiments and Results

4.1 Experimental Setup

Models & Benchmarks. We evaluate Qwen3-1.7B, Qwen3-8B, and Gemma4-12B, covering different model families and scales. For Qwen3 models, we use four reasoning benchmarks: **HMMT**, **GPQA-Diamond (GPQA-D)**, **OlympiadBench-Math**, and **Countdown**. For Gemma4-12B, whose baseline performance on Countdown is near saturation, we replace Countdown with a 200-question subset of **SuperGPQA**, comprising 10-way multiple-choice questions primarily in physics, chemistry, and biology. Dataset details are provided in Appendix D.1.

Baselines. We compare our confidence-dynamics-aware aggregation method against standard majority voting and a range of confidence-based test-time scaling strategies. (1) **Standard** (Wang et al., 2022) performs majority voting over all valid trajectories with uniform weights. (2) **Conf. W.** (Kang et al., 2025) performs confidence-weighted majority voting using the global trajectory confidence score, while (3) **Conf. F.** retains only the top 30% or 50% of trajectories ranked by global confidence before standard majority voting. We further consider their tail-confidence counterparts: (4) **Tail Conf. W.** (Fu et al., 2026b) performs weighted majority voting using confidence aggregated over the tail window of each trajectory, while (5) **Tail Conf. F.** retains the top 30% or 50% of trajectories according to tail confidence before standard majority voting. (6) Finally, our **Joint Confidence-Dynamics Aggregation (Ours)** jointly incorporates tail confidence and tail inter-layer confidence dynamics to filter unreliable trajectories, and subsequently aggregates the remaining predictions using their confidence dynamics as voting weights. All methods use the same sampled trajectory pool and confidence definition, enabling a controlled comparison of selection and aggregation strategies.

Evaluation Details. For each problem, we independently sample $N = 10$ reasoning trajectories using each model's officially recommended decoding hyperparameters (including temperature, top-p, and top-k) to maintain optimal generation quality, as detailed in Appendix D.2. The maximum sequence length is 40,000 tokens for HMMT and 32,768 for other datasets. Trajectories reaching this limit are excluded before aggregation, as truncated paths typically fail to synthesize a coherent final answer. We ensure that all competing methods use the same effective candidate pool. Tail-based confidence and inter-layer confidence dynamics are computed over the final 2,048 generated tokens for Qwen3 models and the final 20% of each trajectory for Gemma4. Unless otherwise specified, we use a retention ratio of $\rho = 60\%$ for Qwen3 models and $\rho = 50\%$ for Gemma4 when independently filtering by confidence and confidence dynamics before taking their intersection. We report final-

Dataset	Standard	Conf. F.		Tail Conf. F.		Joint F.
		30%	50%	30%	50%	50%
HMMT	75.76	78.61	78.97	80.28	79.06	**81.56**
Avg. retained	9.22	2.92	4.68	2.92	4.68	3.43
GPQA-D	75.18	75.84	76.08	74.75	76.62	**77.25**
Avg. retained	9.73	2.97	4.91	2.97	4.91	3.63
Olympiad	84.19	**84.94**	84.64	84.42	84.43	84.81
Avg. retained	9.42	2.92	4.77	2.92	4.77	3.69
SuperGPQA	52.42	55.00	53.80	**55.17**	54.42	55.11
Avg. retained	9.71	2.99	4.92	2.99	4.92	3.59
Overall	71.89	73.60	73.37	73.66	73.63	**74.68**
Avg. retained	9.52	2.95	4.82	2.95	4.82	3.59

Table 2: **Pass@1 of retained trajectories on Gemma4-12B ($N = 10$).** *Avg. retained*: mean trajectories retained per problem. Joint F.: intersection of the top 50% by confidence and confidence dynamics.

Comparison	Trajectory subset	Avg. # / problem	Acc. (%)
vs. Tail C@30%	Retained by baseline	2.92	81.71
	Removed by ours	0.48	62.07
	Added by ours	1.00	**93.33**
vs. Tail C@50%	Retained by baseline	4.68	81.14
	Removed by ours	1.25	62.67
Retained by ours	-	3.43	**87.86**

Table 3: **Retained-set differences across HMMT problems on Gemma4-12B ($N = 10$).** We compare our joint filtering with tail-confidence filtering at retention ratios of 30% and 50%. Removed and added denote trajectories retained **only** by the baseline and only by ours, respectively. We report the average number of trajectories per problem and micro-averaged trajectory accuracy within each subset.

answer accuracy (%) after test-time aggregation. For each model, we additionally report an overall score as the macro-average accuracy across the four evaluated benchmarks for each model.

4.2 Main Results

Table 1 compares our method with standard majority voting and confidence-based baselines under the same trajectory pools. Our method achieves the highest overall accuracy across all three models, outperforming standard majority voting by $+4.24$, $+4.29$, and $+0.92$ percentage points on Qwen3-1.7B, Qwen3-8B, and Gemma4-12B, respectively, while surpassing the strongest confidence-only baseline on the overall metric. Notably, static confidence becomes less effective on the more capable Gemma4-12B, where most confidence-based baselines underperform standard majority voting. For example, 30% confidence filtering reduces overall accuracy from 76.48% to 75.64%, whereas our method improves it to 77.40%. This suggests that inter-layer confidence dynamics provide complementary information when static confidence alone becomes less discriminative.

Across models, confidence-based filtering is generally more effective than confidence-based weighting, suggesting that confidence signals are better suited for identifying and removing unreliable trajectories than for directly calibrating voting weights. This motivates our design choice of joint filtering before weighted aggregation. Furthermore, tail confidence alone does not consistently outperform global confidence. Instead, combining tail confidence with tail inter-layer dynamics yields more robust test-time aggregation.

4.3 Understanding Joint Filtering and Aggregation

We examine how confidence dynamics complements static confidence on Gemma4-12B through retained-trajectory statistics and an illustrative example.

Quality of retained trajectories. Table 2 reports retained-trajectory Pass@1 averaged across problems, with Overall further averaged across benchmarks. Joint filtering achieves the highest overall Pass@1 of 74.68%. It retains more trajectories than tail-confidence filtering at 30% with higher overall accuracy, preserving a broader candidate pool without sacrificing quality. Compared with the 50% baseline, it retains fewer trajectories with higher accuracy. Gains are most evident on HMMT and GPQA-D. On Olympiad and SuperGPQA, its accuracy remains close to the best confidence-only baseline while retaining more trajectories, balancing retained-pool quality and size.

Differences in retained sets. Table 3 summarizes retained-set differences across HMMT problems. Relative to tail-confidence filtering at 30%, our method removes trajectories with 62.07% accuracy and adds trajectories with 93.33% accuracy, improving retained-pool accuracy from 81.71% to 87.86%. Relative to the 50% baseline, it removes less accurate trajectories (62.67%), raising retained-pool accuracy from 81.14% to 87.86%. These results show that dynamics helps exclude unreliable high-confidence trajectories while preserving correct ones missed by a stricter confidence cutoff.

Illustrative example. Figure 4 illustrates how the two signals jointly inform filtering and aggregation. The incorrect trajectories τ_0 and τ_5 confidently commit to 132 without effective correction, despite the pentagon's total area being only 84 as indicated by the problem statement. Their lower dynamics excludes them from the joint set, whereas the correct τ_3, discarded by C@30%, is retained. Within

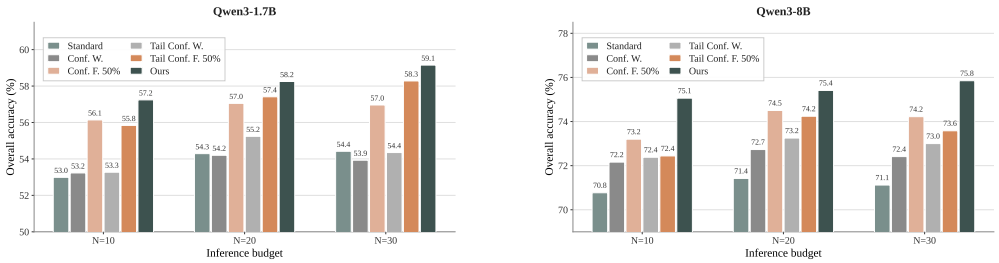


Figure 5: **Performance under increasing test-time generation budgets.** Overall accuracy (%) of different aggregation strategies with $N \in \{10, 20, 30\}$ sampled reasoning trajectories per problem on Qwen3-1.7B (left) and Qwen3-8B (right). All methods share the same generated trajectory pool at each budget. Our confidence-dynamics-aware method consistently achieves the highest overall accuracy as the generation budget increases.

the resulting set $\{\tau_3, \tau_6, \tau_9\}$, the correct trajectories τ_3 and τ_9 also receive larger dynamics-based weights than the incorrect τ_6. Their combined weight is 20.83, versus 9.93 for answer 132, yielding the correct answer 36. Both confidence-only filtering baselines select 132, while full-pool standard majority voting also answers this problem correctly.

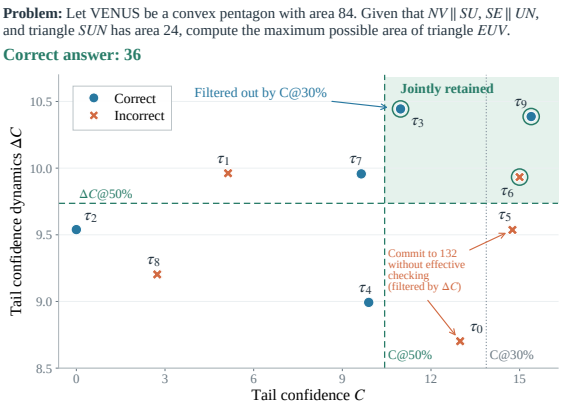


Figure 4: **Trajectory selection on a Gemma4-12B example.** Each point is a trajectory. The shaded region denotes the intersection of the top 50% selected independently by tail confidence C and dynamics ΔC. Joint filtering retains the correct τ_3 missed by C@30%, and removes the incorrect τ_0, τ_5 retained by C@50%.

4.4 Effectiveness over Larger Generation Budgets

We further evaluate whether our method remains effective as the test-time generation budget increases. We vary the number of sampled trajectories as $N \in \{10, 20, 30\}$ and report the macro-average accuracy across the four benchmarks. Results for Gemma4-12B are provided in Appendix E.

As shown in Figure 5, our method achieves the highest overall accuracy across all evaluated sampling budgets on both Qwen3-1.7B and Qwen3-8B, with accuracy increasing as more trajectories are sampled. The Gemma4-12B results in Appendix E show that our method also maintains its advantage across budgets, although its accuracy does not increase monotonically. These results demonstrate that the benefit of incorporating inter-layer confidence dynamics for joint aggregation generalizes across different test-time compute budgets.

4.5 Varying Joint Aggregation Strategies

We ablate two key components of our aggregation strategy under an inference budget of $N = 10$: the retention ratio for joint filtering and the weighting scheme after filtering. We vary ρ from 30% to 70% and compare unweighted, confidence-weighted, dynamics-weighted, and joint-weighted voting, with standard majority voting and confidence-only filtering as references.

As shown in Figure 6, joint confidence-dynamics filtering consistently outperforms standard majority voting across a broad range of retention ratios. The preferred retention ratio varies with model scale: Qwen3-1.7B favors a larger retention ratio around 60%, whereas Qwen3-8B and Gemma4-12B favor more selective filtering around $\rho = 40\% \sim 50\%$. Overall, $\rho = 50\%$ provides a robust general-purpose setting across models. Similar trends under larger inference budgets are reported in Appendix F. The weighting strategy after filtering further affects performance. Across retention ratios, dynamics-weighted or joint-weighted voting generally outperforms unweighted and confidence-only weighting, showing that ΔC is useful for both trajectory selection and answer aggregation. We therefore adopt joint confidence-dynamics filtering followed by dynamics-weighted voting.

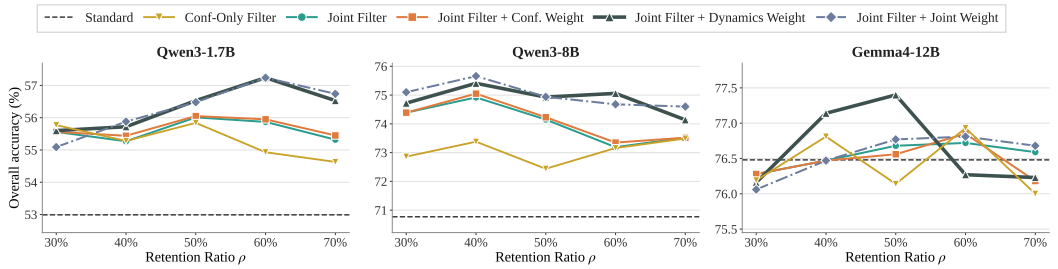


Figure 6: **Ablation of joint filtering and weighting strategies.** Overall accuracy (%) under varied retention ratios ρ on three models. We compare confidence-only filtering, joint confidence-dynamics filtering, and different weighting strategies applied after joint filtering. Dashed lines denote standard majority voting without filtering.

4.6 Confidence Dynamics across Different Layer Depths

Our default confidence dynamics compare the final and penultimate layers to capture localized late-stage updating. We further examine how this signal changes across layer depths using $L-1$, $L-2$, or $L-3$ as the reference layer, computing $\Delta C = C^{(L)} - C^{(\ell)}$ within the same joint aggregation framework. We evaluate Qwen3-1.7B and Qwen3-8B with $N = 10$ on HMMT, GPQA-Diamond, and Countdown across different retention ratios, using the final 2,048 tokens as the tail window.

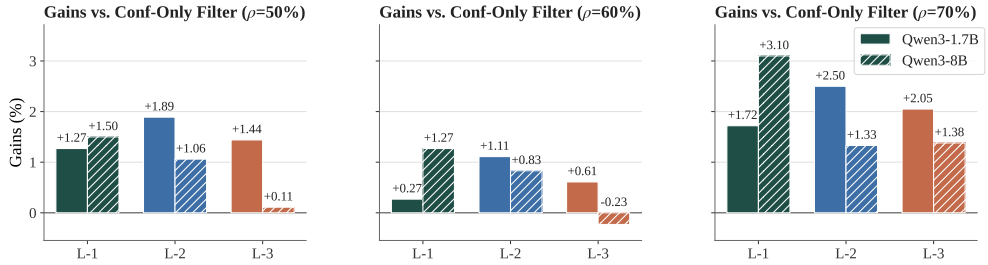


Figure 7: **Confidence dynamics across different layer depths.** Average accuracy gains over confidence-only filtering when computing confidence dynamics between the final layer L and layer $L-1$, $L-2$, or $L-3$, averaged over HMMT, GPQA-Diamond, and Countdown with $N = 10$. Results are shown for Qwen3-1.7B and Qwen3-8B under retention ratios $\rho \in \{50\%, 60\%, 70\%\}$.

As shown in Figure 7, confidence dynamics using either $L-1$ or $L-2$ consistently improve over confidence-only filtering across varied evaluated settings, whereas gains from $L-3$ are less consistent. We hypothesize that representations closer to the final readout are better aligned with the output space, making their confidence change more informative about late-stage prediction updating. Earlier reference layers may introduce greater readout mismatch, reducing the reliability of this signal. We therefore adopt the penultimate layer ($L-1$) as a broadly effective reference in our main method.

5 Related Works and Concluding Remarks

Test-time scaling improves reasoning through sampling and aggregation. Self-consistency uses majority voting (Wang et al., 2022), while subsequent work explores Best-of-N and verifier-guided selection (Brown et al., 2024; Snell et al., 2024; Lightman et al., 2024). Recent approaches use model-internal confidence for verifier-free weighting and filtering (Taubenfeld et al., 2025; Kang et al., 2025; Fu et al., 2026b). However, high confidence can coexist with incorrect reasoning and hypothesis lock-in (Wan et al., 2025; Hiremath & Hiremath, 2026). Unlike approaches studying entropy dynamics across generation steps (Zhu et al., 2026), we measure confidence changes across internal layers to complement static confidence. Appendix B provides a fuller discussion.

Concluding Remarks. Static confidence alone provides limited discrimination among high-confidence reasoning trajectories. We introduce inter-layer confidence dynamics as a complementary signal and combine it with confidence for joint filtering and weighted aggregation. Across three models, this training-free approach improves overall reasoning accuracy over majority voting and confidence-based baselines. Our analyses associate stronger confidence dynamics with higher accuracy and less observed stagnation, highlighting their value for test-time reasoning.

AI USE STATEMENT

In this work, we used generative AI tools to assist with paper writing and presentation, including language editing, organizing experimental results, checking for writing and formatting issues, and brainstorming preliminary figure layouts. We additionally used generative AI as a discussion tool to obtain feedback on research methodology and experimental design. All AI-assisted suggestions were critically reviewed by the authors, and methodological or experimental suggestions were incorporated only after further analysis or empirical validation. Generative AI was not used to generate experimental results or determine the scientific conclusions of this work. We reviewed and verified all AI-assisted content for accuracy and consistency with our experiments, and take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We define the token-level confidence and inter-layer confidence dynamics measures, their trajectory-level aggregation, and the joint filtering and voting procedure in Sections 2 and 3. Section 4.1 specifies the evaluated models and benchmarks, sampling budgets, tail windows, retention ratios, and treatment of truncated trajectories. All aggregation methods are evaluated on the same generated trajectory pool for each problem, allowing direct comparison under identical sampling conditions. The trajectory segmentation procedure, behavioral label definitions, and judge prompt used in our stagnation analysis are provided in Appendix C, while model-specific decoding settings and maximum generation lengths are detailed in Appendix D.2. Additional results across sampling budgets and aggregation settings are reported in Appendices E and F. Upon acceptance, we will publicly release: (i) our complete codebase, including trajectory generation, confidence and inter-layer dynamics computation, filtering, voting, and evaluation scripts; (ii) the constructed Countdown dataset; and (iii) the selected SuperGPQA evaluation subset, enabling all reported results to be fully reproduced from the released artifacts.

REFERENCES

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2025. URL https://arxiv.org/abs/2303.08112.

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.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. *arXiv preprint arXiv:2605.00674*, 2026.

Xeron Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *Advances in Neural Information Processing Systems*, 38, 2025.

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

Tairan Fu, Javier Conde, Gonzalo Martinez, Maria Grandury, and Pedro Reviriego. Multiple choice questions: Reasoning makes large language models (llms) more self-confident, specially when they are wrong. *IEEE Intelligent Systems*, 2026a.

Yichao Fu, Xuewei Wang, Hao Zhang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. In *International Conference on Learning Representations*, volume 2026, pp. 94355–94377, 2026b.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Prakul Sunil Hiremath and Harshit R Hiremath. Calibration drift under reasoning: How chain-of-thought budgets induce overconfidence in large language models. *arXiv preprint arXiv:2606.11211*, 2026.

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. *Advances in neural information processing systems*, 38:19720–19745, 2025.

Lecheng Kong, Like Hui, Haitao Mao, and Jun Huan. Consilience for verifier-free test-time scaling. *arXiv preprint arXiv:2608.09898*, 2026.

Jiakun Li, Xingwei He, Kefan Li, Hongzheng Chai, Hongyue Yu, and Yuan Yuan. Efficient test-time scaling via temporal reasoning aggregation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 13304–13318, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

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. *arXiv preprint arXiv:2311.12022*, 2023.

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.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Yue Wan, Xiaowei Jia, and Xiang Lorraine Li. Unveiling confirmation bias in chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3788–3804, 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.

Yu Wang, Minghao Liu, Jiayun Wang, Jinrui Huang, Ankit Shah, and Wei Wei. Inference time optimization with confidence dynamics. *arXiv preprint arXiv:2605.25244*, 2026.

Zhenyu Wang. Logitlens4llms: Extending logit lens analysis to modern large language models. *arXiv preprint arXiv:2503.11667*, 2025.

Shangyu Xing, Siyuan Wang, Chenyuan Yang, Xin-Yu Dai, and Xiang Ren. Lookahead tree-based rollouts for enhanced trajectory-level exploration in reinforcement learning with verifiable rewards. In *International Conference on Learning Representations*, volume 2026, pp. 31759–31776, 2026.

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.

Chenghua Zhu, Siyan Wu, Xiangkang Zeng, Zishan Xu, Zhaolu Kang, Yifu Guo, Yuquan Lu, Junduan Huang, and Guojing Zhou. Edis: Diagnosing llm reasoning via entropy dynamics. *arXiv preprint arXiv:2602.01288*, 2026.

Appendix

Table of Contents

A Detailed Dynamics of the Motivation Example **13**

B Extended Related Work **13**

C Reasoning Behavior Annotation **14**
C.1 Trajectory Segmentation 14
C.2 Segment-Level Functional Classification 14
C.3 Stagnation Ratio 14
C.4 Judge Prompt 15

D Additional Experimental Setups **15**
D.1 Benchmark Details 15
D.2 Sampling Hyperparameters 16

E Additional Results on Larger Generation Budgets **16**

F Joint Aggregation Ablation under Larger Inference Budget **16**

A Detailed Dynamics of the Motivation Example

We further examine the two Qwen3-8B trajectories in Figure 1 at the token level. Figure 8 shows their confidence C and inter-layer confidence dynamics ΔC throughout generation. For visualization, we report the raw token-level measurements together with a rolling mean using a window of 256 tokens.

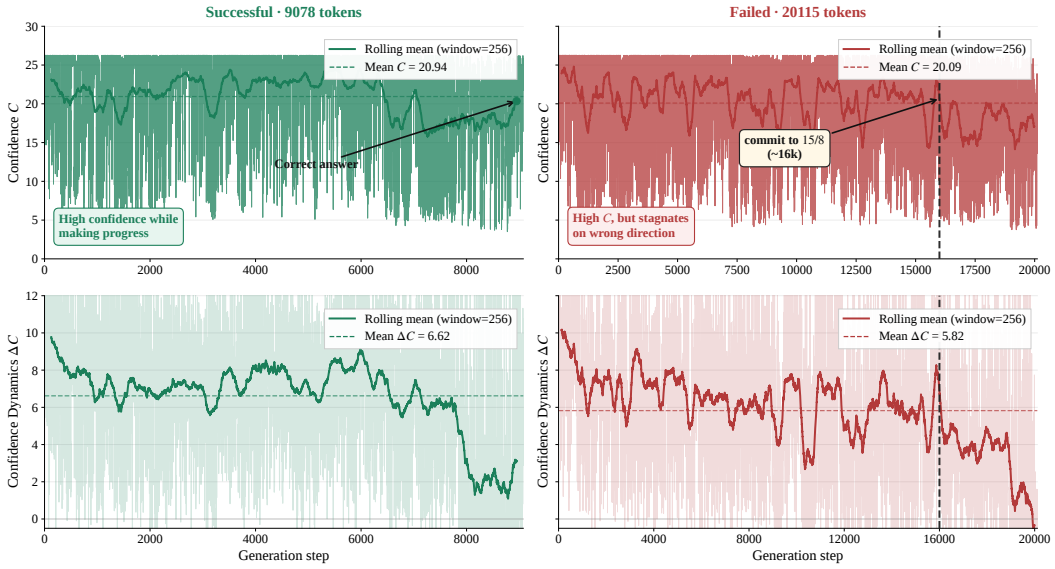


Figure 8: **Token-level confidence and confidence dynamics for the trajectories in Figure 1.** The left column shows the successful trajectory (9,078 tokens), and the right column shows the failed trajectory (20,115 tokens). Top: final-layer confidence C throughout generation. Bottom: inter-layer confidence dynamics ΔC between the penultimate and final layers. Solid curves denote rolling means with a 256-token window, and dashed horizontal lines indicate trajectory-level means. For the failed trajectory, the vertical dashed line marks the region where the reasoning commits to the incorrect $15/8$ conjecture and subsequently stagnates.

As shown in Figure 8, static confidence provides limited separation as both trajectories remain highly confident throughout most of generation, including their tail portions, despite producing different answers. In contrast, their confidence dynamics exhibit distinct patterns. The failed trajectory remains actively updated early on, but around 16K tokens it commits to the incorrect $15/8$ conjecture and subsequently reasons around the same direction with little effective correction.

Notably, after entering this stagnant range, its ΔC drops sharply throughout the tail and approaches zero near the end, while its confidence remains high. The successful trajectory, in comparison, maintains stronger confidence dynamics over most of its reasoning process while continuing to make progress toward the correct solution. This contrast illustrates how ΔC, particularly in the tail, provides complementary information for identifying high-confidence reasoning stagnation.

B Extended Related Work

Test-time scaling improves LM reasoning by sampling and aggregating multiple reasoning trajectories. Self-consistency selects the most frequent answer through majority voting (Wang et al., 2022), while Best-of-N, verifier-guided selection, and reward models aim to identify reliable solutions (Brown et al., 2024; Snell et al., 2024; Lightman et al., 2024). Verifier-free approaches instead use model-internal signals: Confidence-Informed Self-Consistency weights votes by model confidence (Taubenfeld et al., 2025); Kang et al. (2025) propose *self-certainty* for Best-of-N selection and combine confidence-based ranking with answer frequency; and DeepConf filters trajectories using token-level confidence (Fu et al., 2026b).

Confidence, however, does not always track reasoning quality. Reasoning can increase confidence even for incorrect answers (Fu et al., 2026a), and confirmation bias or hypothesis lock-in can reinforce an incorrect belief instead of correcting it (Wan et al., 2025; Hiremath & Hiremath, 2026). Similarly,

uniformly high confidence may reflect insufficient exploration (Kong et al., 2026). These findings relate to our *high-confidence reasoning stagnation* that a trajectory remains confident while making limited effective progress and correction.

To capture information beyond static confidence, recent work examines its evolution *across generation steps*. Confidence gains from head to tail windows have been used to weight votes (Wang et al., 2026), entropy dynamics to select trajectories (Zhu et al., 2026), early-to-final confidence changes to assess exploration and convergence (Kong et al., 2026), and confidence trajectories together with answer consistency to decide when to stop generation (Li et al., 2026). A separate line of work, including the Logit Lens and Tuned Lens, reads intermediate hidden states to study how token predictions evolve *across model layers* (Wang, 2025; Belrose et al., 2025). Motivated by this layerwise perspective, we measure the confidence change from the penultimate to the final layer at each token and aggregate it over the reasoning trajectory. We combine this signal with tail confidence for trajectory filtering and weighted aggregation.

C Reasoning Behavior Annotation

To investigate the behavioral patterns associated with inter-layer confidence dynamics, we perform a segment-level analysis of reasoning trajectories. Our goal is to quantify whether trajectories with weak confidence dynamics exhibit less effective reasoning progress and more stagnation, rather than simply generating longer outputs.

C.1 Trajectory Segmentation

We segment each reasoning trajectory using a sentence-aware procedure. We first split the generated reasoning into sentences according to period boundaries. We then sequentially merge adjacent sentences until the resulting segment contains at least 500 tokens, after which a new segment is started. The final residual segment is retained even if it contains fewer than 500 tokens.

This procedure preserves natural sentence boundaries while producing segments with sufficient context for functional reasoning analysis. The same segmentation procedure is applied to all trajectories, independent of their correctness, confidence, or confidence dynamics.

C.2 Segment-Level Functional Classification

We use Gemma4-12B as a judge to characterize the primary reasoning function of each segment. For each segment s_i, the judge is provided with the problem, the preceding reasoning context, and the current segment, and assigns exactly one of the following four labels:

- **Progress**: The segment introduces a new intermediate result, constraint, deduction, or other information that substantively advances the current solution state.
- **Verification**: The segment substantively checks an existing result or reasoning step and provides new evidence that supports or challenges it.
- **Revision / Strategy Switch**: The segment revises a previous intermediate conclusion, abandons an existing reasoning direction, or substantially changes the solution strategy.
- **Stagnation**: The segment primarily repeats, restates, or continues existing reasoning without introducing a new effective deduction, verification, correction, or change in solution direction.

These categories distinguish effective reasoning updates, including progress, verification, and revision from stagnation that continues without substantial reasoning progress. Importantly, the judge is not provided with trajectory correctness, confidence scores, or inter-layer confidence dynamics during annotation, preventing these signals from directly influencing the behavioral labels.

C.3 Stagnation Ratio

For a trajectory τ containing M annotated reasoning segments, we define its *stagnation ratio* as

$$R_{\text{stag}}(\tau) = \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}\left[\text{label}(s_i) = \text{Stagnation}\right]. \tag{11}$$

A higher R_{stag} therefore indicates that a larger fraction of the generated reasoning is spent without introducing an effective reasoning update.

For the analysis in Figure 3, we first restrict the trajectory pool to the top-50% samples ranked by global confidence, excluding trajectories truncated at the maximum generation length. We then divide the resulting high-confidence subset into three equal-sized groups—*Low*, *Mid*, and *High*—according to their global inter-layer confidence dynamics ΔC. For each group, we report the average stagnation ratio, final-answer accuracy, and mean sequence length.

C.4 Judge Prompt

We instruct the judge to classify the functional role of each reasoning segment relative to the preceding reasoning context when provided, rather than evaluating its writing quality or predicting whether the final answer is correct. The prompt follows the template below:

> **Prompt for Segment-Level Reasoning Classification**
>
> You are labeling contiguous segments of a math/reasoning trajectory. Given a reasoning segment (and optional preceding context), choose **exactly one** of the following labels:
>
> 1. **Progress:** Derives new intermediate results, constraints, or facts that substantively advance the current solution state. *Core: effective advancement along the current direction.*
> 2. **Verification:** Substantively checks existing results or reasoning and produces new supporting or refuting evidence. *Core: effective checking.*
> 3. **Revision / Switch:** Modifies an existing intermediate conclusion, abandons the current direction, or substantially changes the solution strategy. *Core: effective correction or change of direction.*
> 4. **Stagnation:** Repeats, restates, or continues existing reasoning without introducing new effective information, deduction, or correction. *Core: no effective advancement.*
>
> **Rules**
>
> - Judge only the provided segment and the preceding context.
> - Prefer **Stagnation** when the segment primarily repeats previous reasoning without introducing substantive new content.
> - Use **Revision / Switch** only when the strategy or an intermediate conclusion clearly changes.
> - Output only one of the following label strings, with no additional explanation: Progress; Verification; Revision / Switch; Stagnation

D Additional Experimental Setups

D.1 Benchmark Details

We evaluate both Qwen3 models on HMMT, GPQA-Diamond, OlympiadBench-Math, and Countdown. For Gemma4-12B, we replace Countdown with SuperGPQA due to near-saturated baseline performance on Countdown.

HMMT contains 60 problems from the February and November 2025 Harvard–MIT Mathematics Tournament competitions. **GPQA-Diamond** comprises 198 expert-level multiple-choice questions in physics, chemistry, and biology. For **OlympiadBench-Math**, we use a 675-problem subset evaluating olympiad-level mathematical reasoning.

We construct 300 **Countdown** problems, each providing six numbers and a target. The model must select two numbers from the first three and two from the last three, using each selected number exactly once to form an expression equal to the target. For **SuperGPQA**, we select 200 questions with ten answer choices each, primarily covering physics, chemistry, and biology. We will release the

Countdown data and construction and filtering scripts, together with the SuperGPQA subset and its selection procedure.

D.2 Sampling Hyperparameters

For each problem, we independently sample N candidate trajectories from the target model. Unless otherwise specified (e.g., inference-budget sweeps with $N \in \{10, 20, 30\}$), we use a default budget of $N = 10$. We generally follow the recommended decoding configuration of each model family and set the maximum generation length according to the benchmark. Across all runs, we use $\text{top-p} = 0.95$, $\text{min-p} = 0$.

Qwen3-1.7B and Qwen3-8B. We follow the recommended Qwen3 decoding configuration with $\text{temperature} = 0.6$, $\text{top-k} = 20$, and $\text{top-p} = 0.95$. The only exception is Countdown, for which we increase the temperature to 1.0 while keeping the remaining sampling parameters unchanged. Unlike conventional question answering, Countdown requires combinatorial exploration over different number selections, operation orders, and arithmetic constructions under explicit constraints. We therefore use a higher sampling temperature to encourage broader exploration of diverse solution paths rather than repeatedly sampling similar reasoning trajectories. Maximum new tokens are $40{,}000$ for HMMT and $32{,}768$ for GPQA-Diamond, OlympiadBench-Math, and Countdown.

Gemma4-12B. We follow the recommended Gemma 4 decoding configuration with $\text{temperature} = 1.0$, $\text{top-k} = 64$, and $\text{top-p} = 0.95$, together with a presence penalty of 0 and a repetition penalty of 1.0. Maximum new tokens are $40{,}000$ for HMMT and $32{,}768$ for GPQA-Diamond, OlympiadBench-Math, and SuperGPQA. Countdown is not included for Gemma4-12B.

Model	Benchmark	T	top-k	top-p	max tokens
Qwen3-1.7B / 8B	HMMT	0.6	20	0.95	40,000
	GPQA-Diamond	0.6	20	0.95	32,768
	OlympiadBench-Math	0.6	20	0.95	32,768
	Countdown	1.0	20	0.95	32,768
Gemma4-12B	HMMT	1.0	64	0.95	40,000
	GPQA-Diamond	1.0	64	0.95	32,768
	OlympiadBench-Math	1.0	64	0.95	32,768
	SuperGPQA	1.0	64	0.95	32,768

Table 4: Decoding hyperparameters used for trajectory sampling.

E Additional Results on Larger Generation Budgets

We provide additional generation-budget scaling results for Gemma4-12B. Following the same evaluation protocol, we vary the number of sampled trajectories as $N \in \{10, 15, 20\}$, with all aggregation methods operating on the same trajectory pool at each budget.

As shown in Figure 9, our method consistently achieves the best overall performance across all three generation budgets, reaching 77.4%, 78.8%, and 77.6% at $N = 10$, 15, and 20, respectively. In contrast, simply increasing the sampling budget does not lead to monotonic improvements for standard majority voting or confidence-only aggregation. Notably, while all methods benefit from increasing the budget from $N = 10$ to $N = 15$, their performance generally decreases at $N = 20$, yet our method maintains a clear advantage over the baselines. Together with the Qwen3 results in Figure 5, these results further demonstrate that confidence-dynamics-aware aggregation remains effective across model families and different test-time sampling budgets.

F Joint Aggregation Ablation under Larger Inference Budget

We further examine the robustness of our joint filtering and weighting strategies under a larger inference budget of $N = 20$. Following the same protocol as Section 4.5, we vary the retention ratio ρ from 30% to 70% and compare confidence-only filtering with joint confidence-dynamics filtering under different weighting strategies.

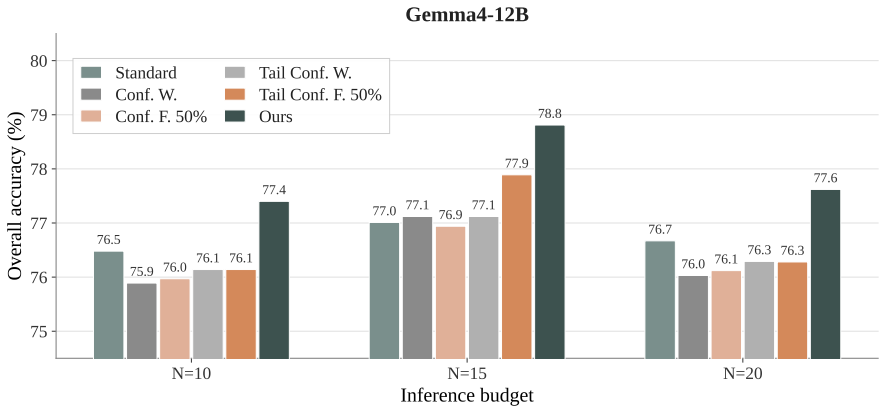


Figure 9: **Performance under increasing test-time generation budgets on Gemma4-12B.** Overall accuracy (%) of different aggregation strategies with $N \in \{10, 15, 20\}$ sampled trajectories per problem. Our method consistently achieves the highest overall accuracy across all generation budgets.

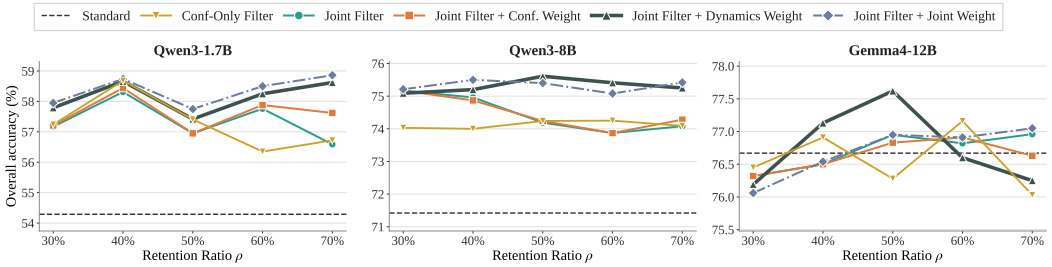


Figure 10: **Ablation of joint filtering and weighting strategies under a larger inference budget** ($N = 20$)**.** Overall accuracy (%) across different retention ratios ρ on Qwen3-1.7B, Qwen3-8B, and Gemma4-12B. We compare confidence-only filtering with joint confidence-dynamics filtering followed by unweighted, confidence-weighted, dynamics-weighted, or joint-weighted voting. The dashed line denotes standard majority voting without filtering.

As shown in Figure 10, the trends observed at $N = 10$ remain largely consistent under the larger inference budget. Joint confidence–dynamics filtering is effective across a broad range of retention ratios and generally outperforms confidence-only filtering. The optimal retention ratio remains model dependent: Qwen3-1.7B benefits from relatively larger retention ratios, whereas Qwen3-8B and Gemma4-12B favor more selective filtering in several settings. Nevertheless, intermediate retention ratios provide robust performance across models.

Dynamics-aware weighting also remains consistently competitive after joint filtering. In particular, dynamics-based and joint weighting generally outperform or closely match unweighted and confidence-only weighting across retention ratios. These results confirm that the effectiveness of joint filtering and dynamics-aware aggregation is not specific to the $N = 10$ setting, but persists when the candidate pool is enlarged.