ConMA : Confidence-Guided Kernel Sampling with Multi-Stage Aggregation for LLM Reasoning

Anonymous ACL submission

Abstract

Test-time scaling (TTS) enhances LLM reasoning capabilities by sampling and aggregating diverse solution trajectories. However, existing approaches often rely on external verifiers and one-shot independent sampling, which results in inefficient budget allocation and underutilizes interim high-quality trajectories. We propose ConMA, a training-free, verifier-free TTS framework that reallocates a fixed inference budget into iterative *sample–filter–diversify–select* cycles: it filters answer groups based on intrinsic token-probability confidence, enriches candidates through diversity-aware expansion, and employs repeated single-choice selection for multi-stage refinement. Across multiple benchmarks, ConMA consistently improves accuracy under fixed budgets. With a maximum budget of $N = 64$, ConMA boosts Qwen3-4B to 80% accuracy on AIME25, significantly outperforming strong baselines while converging early with only 18 samples on average, substantially reducing inference cost.

1 Introduction

The rapid progress of Large Language Models (LLMs) is increasingly driven not only by scaling model capacity, but also by allocating more computation at inference time (Brown et al., 2020; OpenAI et al., 2024). While Chain-of-Thought (CoT) prompting (Wei et al., 2022; Kojima et al., 2022) unlocks strong reasoning behavior, recent studies show that test-time scaling (TTS)—sampling and aggregating multiple reasoning trajectories within an inference budget—can match the performance of models at a significantly larger scale (Wang, 2025; Snell et al., 2025). Accordingly, majority voting (Wang et al., 2023), Best-of-N selection (Cobbe et al., 2021), and search-based decoding (Yao et al., 2024; Zhenting Qi, 2025) have become standard protocols for challenging mathematical and logical tasks.

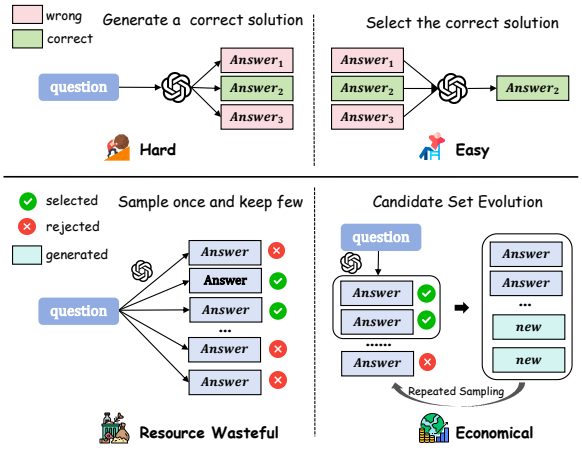

Figure 1: **Motivating Candidate Set Evolution.** While generating correct solutions is harder than selecting them, standard methods wastefully generate large batches in one shot. Our approach maximizes efficiency under a fixed budget by iteratively pruning low-quality paths, generating diverse explorations from survivors, and leveraging the model's intrinsic selection capability to refine the candidate set for the next round.

Current TTS frameworks largely rely on advanced search strategies augmented by external verifiers (Lightman et al., 2024; Wang et al., 2024; Uesato et al., 2023) or mainstream sampling-based approaches (e.g., Best-of-N and self-consistency) that operate under a one-shot large-batch protocol (Wang et al., 2023; Snell et al., 2025). Despite their empirical success, this prevailing paradigm faces structural limitations regarding scalability and efficiency. **First**, the dependence on auxiliary Reward Models necessitates massive amounts of fine-grained supervision and imposes substantial deployment and inference overhead (Wang, 2025). **Second**, the homogeneous nature of independent trajectory generation fails to reuse high-quality intermediate reasoning traces(Li et al., 2023; Hong et al., 2025). Consequently, the effective search space is governed by unguided randomness, leading to redundant or narrowly clustered solutions.

Third, the difficulty-agnostic strategy—*sample once and keep few*—misallocates computation on low-quality candidates that are inevitably discarded. This rigidity leads to overshooting budgets on simple tasks while hitting performance plateaus on complex problems compared to iterative refinement.

To bridge these gaps, we introduce **ConMA** (**Con**fidence-guided **M**ulti-stage **A**ggregation), a training-free, data-free, and plug-and-play TTS framework. Unlike methods that depend on external supervision, ConMA repurposes the LLM's own internal uncertainty as a guidance signal to orchestrate an adaptive reasoning loop. The framework operates through three synergistic mechanisms: (1) **Confidence-Guided Kernel Sampling (CKS)** uses token-probability–based confidence to score trajectories and perform answer-group filtering, retaining a compact "kernel" of high-quality candidate groups; (2) **Diversity-Seeking Exploration (DSE)** breaks the homogeneous sampling paradigm by conditioning on kernel representatives to generate additional trajectories with deliberately different reasoning patterns, thereby reusing high-quality intermediate traces while expanding the search space; and (3) **Multi-Stage Aggregation (MSA)** is a budget-preserving outer loop inspired by Figure 1: instead of one-shot *sample once and keep few*, it partitions a predefined maximum budget N into small-batch rounds that iteratively refine the search space by pruning weak candidates and expanding from high-quality survivors. Each round applies CKS to prune low-quality answer groups early, uses DSE to expand promising representatives, and then converts updating the sample pool into repeated *single-answer* multiple-choice selection; the n selected trajectories are recycled as inputs for the subsequent round, driving a process of progressive refinement.

Our approach balances reasoning accuracy and computational cost via a convergence-aware multi-stage loop with adaptive exploration. Across four benchmarks, ConMA consistently outperforms strong baselines. On the challenging AIME25 dataset, ConMA enables Qwen3-4B to achieve 80% accuracy. With convergence-based early stopping (and adaptive DSE supplementation), ConMA reaches this performance with only 18 samplings on average under the predefined maximum budget of $N = 64$, substantially reducing inference cost compared to uniform scaling strategies.

In summary, our contributions are as follows:

- We propose ConMA, a training-free and verifier-free TTS framework that reallocates inference budget into iterative *sample–filter–diversify–select* cycles for LLM mathematical reasoning.
- We develop confidence-guided answer-group filtering, survivor-conditioned diversity-seeking exploration, and a multi-stage resampling scheme via repeated single-answer selection with convergence-based early stopping.
- Experiments on four benchmarks show consistent gains over strong baselines; ConMA achieves 80% on AIME25 with Qwen3-4B while using only 18 samplings on average under $N = 64$.

2 Related Work

2.1 Sampling-based test-time scaling and aggregation

A dominant line of test-time scaling (TTS) improves LLM reasoning by sampling multiple Chain-of-Thought trajectories and aggregating their outcomes, including self-consistency and majority voting (Wang et al., 2023) as well as Best-of-N selection. These approaches are effective by increasing the chance of generating a correct trajectory(Snell et al., 2024), but they typically follow a one-shot paradigm: trajectories are sampled independently and only aggregated or filtered after the full budget has been spent. This "sample once and keep few" workflow can be sample-inefficient and does not leverage information contained in earlier high-quality trajectories to guide subsequent exploration(Li et al., 2023). In contrast, ConMA reallocates a fixed budget into round-based cycles that filter early and grow from surviving hypotheses.

2.2 Verifier- and reward-guided reranking

To better distinguish high-quality reasoning traces, many methods augment TTS with external verifiers or reward models that score trajectories for reranking, selection, or weighted aggregation (Lightman et al., 2024; Wang et al., 2024). Such outcome- or process-level reward models can substantially improve performance, especially on hard reasoning tasks, but they require additional supervision to train and add non-trivial deployment and inference overhead. A separate but related direction

studies calibration and uncertainty signals from LLMs, showing that token-level probabilities can correlate with correctness and hallucination (Kang et al., 2025; Yichao Fu, 2026). ConMA builds on this insight by using intrinsic token-probability–based confidence for answer-group filtering, enabling verifier-free selection while preserving a lightweight, training-free test-time pipeline.

2.3 Search and iterative refinement at inference time

Beyond independent sampling, search-style decoding explicitly explores a structured space of partial solutions, such as tree-based reasoning and other guided search procedures (Yao et al., 2024; Zhenting Qi, 2025). In parallel, iterative refinement and diversity-oriented prompting aim to utilize intermediate results to steer subsequent generation, often through feedback, critique, or alternative reasoning attempts(Madaan et al., 2023; Shinn et al., 2023). While these approaches improve exploration, they may require maintaining explicit search structures or performing expensive iterative generation without principled early filtering. ConMA adopts an implicit candidate-set evolution approach: it bypasses the potential short-sightedness of step-wise search by employing global, result-based pruning. By combining answer-group filtering with survivor-conditioned expansion, it effectively exploits the generation–selection asymmetry to refine solutions within a fixed budget.

3 Methodology

In this section, we propose ConMA, a trajectory-level TTS framework(Figure 2). ConMA improves reasoning via a synergistic loop: Confidence-Guided Kernel Sampling (CKS) first filters low-quality answers; Diversity-Seeking Exploration (DSE) then expands survivors with diverse paths; and Multi-Stage Aggregation (MSA) selects the best candidates to initialize the next round. This process iteratively purifies the solution space.

3.1 Confidence-Guided Kernel Sampling

The primary objective of CKS is to effectively prune low-quality reasoning paths from the solution space, thereby identifying a high-quality "Kernel" of candidate trajectories. Distinct from conventional selection paradigms that rely either exclusively on output frequency(Wang et al., 2023) or necessitate computationally expensive external reward models(Wang et al., 2024) for scoring. By harmonizing the model's internal uncertainty signals with external sample consistency, CKS provides a dual-perspective assessment spanning from trajectory-level confidence to group-level reliability.

3.1.1 Single Trajectory Confidence Scoring

Formally, given a query x, the model generates a reasoning trajectory $\tau = \{y_1, y_2, \ldots, y_L\}$ consisting of L tokens. We define the atomic confidence of a sequence as the geometric mean of its token probabilities (Kang et al., 2025):

$$\mathcal{C}(\cdot) = \exp\left(\frac{1}{|\cdot|}\sum_{y_i \in \cdot} \log P(y_i \mid x, y_{<i})\right) \tag{1}$$

Adopting a fixed-window approach to capture uncertainty, we define the composite score $S(\tau)$ as a weighted sum of the full sequence (τ_{full}) and the final W tokens (τ_{tail}):

$$S(\tau) = \alpha \cdot \mathcal{C}(\tau_{\text{full}}) + \beta \cdot \mathcal{C}(\tau_{\text{tail}}) \tag{2}$$

This normalization mitigates the length bias inherent in raw joint probabilities, ensuring that longer, more detailed reasoning chains are not unfairly penalized compared to shorter alternatives.

3.1.2 Group Confidence Calculation

After sampling N trajectories, we aggregate them into semantic groups $\mathcal{G} = \{G_1, \ldots, G_K\}$ based on final answers. To evaluate reliability, we calculate a Composite Group Confidence $S(G_k)$ that combines voting consensus and average intrinsic quality:

$$S(G_k) = \underbrace{\lambda \cdot \frac{|G_k|}{N}}_{\text{Consensus}} + \underbrace{(1-\lambda) \cdot \frac{1}{|G_k|}\sum_{\tau \in G_k} S(\tau)}_{\text{Average Path Quality}} \tag{3}$$

where λ regulates the trade-off between popularity and model certainty, providing a robust metric for kernel pruning.

3.1.3 Kernel Construction via Cumulative Filtering

To define the pruning Kernel, we transform the raw group scores into a probability distribution via a temperature-scaled Softmax:

$$P(G_k) = \frac{\exp(S(G_k)/\tau_{\text{temp}})}{\sum_{j=1}^{K} \exp(S(G_j)/\tau_{\text{temp}})} \tag{4}$$

We then rank the groups in descending order such that $P(G_{(1)}) \geq P(G_{(2)}) \geq \cdots \geq P(G_{(K)})$.

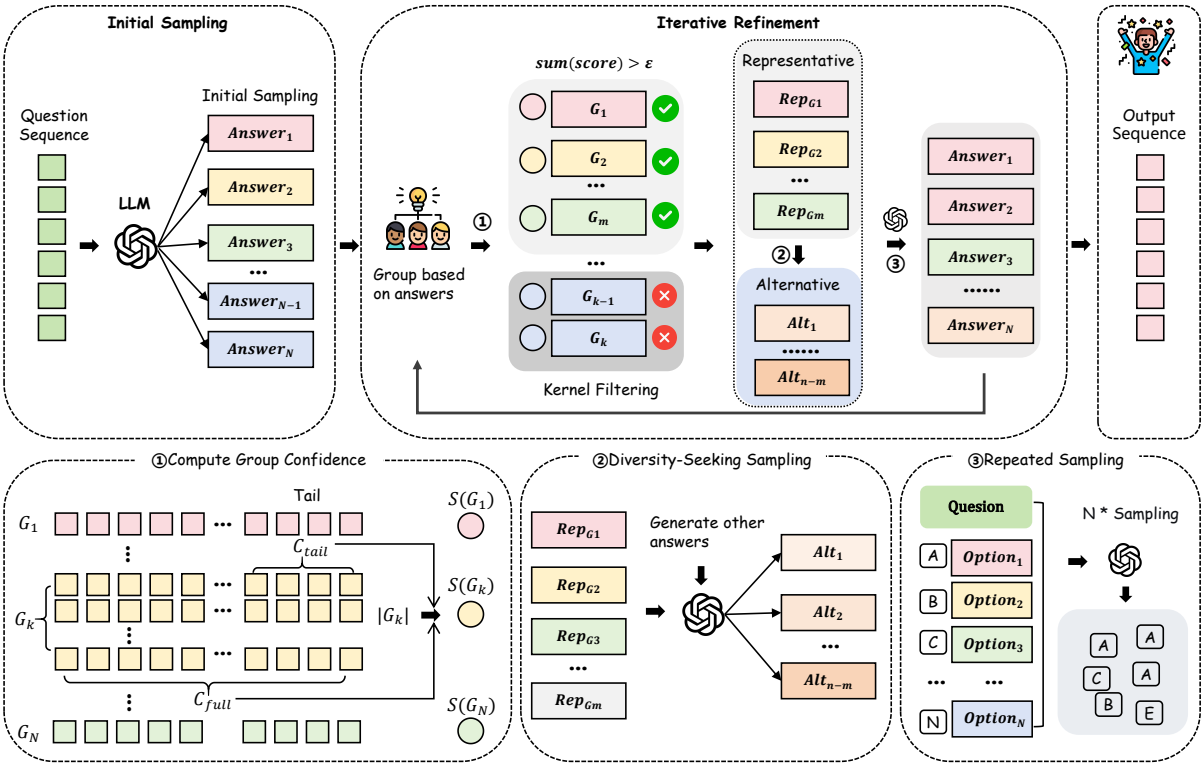

Figure 2: **Overview of the ConMA framework.** Following initial sampling, the pipeline iterates through a synergistic loop: ① **Confidence-Guided Kernel Sampling (CKS)** prunes low-quality answer groups via composite confidence scoring; ② **Diversity-Seeking Exploration (DSE)** expands the search space by generating diverse paths from kernel representatives; and ③ **Multi-Stage Aggregation (MSA)** consolidates consensus through discriminative MCQ selection.

We construct the Kernel $\mathcal{K}_t$ using cumulative probability thresholding. We retain the smallest set of top-ranked groups whose summed probabilities satisfy a threshold ρ:

$$\sum_{i=1}^{m} P(G_{(i)}) \geq \rho \tag{5}$$

This strategy dynamically filters out the long tail of low-quality candidates while preserving the most plausible answer clusters.

3.2 Diversity-Seeking Exploration (DSE)

The DSE module expands and refines the search space by conditioning on the high-quality kernels $\mathcal{K}_t$ established by CKS. We first select a representative $\hat{\tau}_k = \arg\max_{\tau \in G_k} S(\tau)$ from each group G_k to form the set $\mathcal{R}_t$. These representatives are integrated with the original query Q into a prompt $\mathcal{T}_{\text{DSE}}$ (see Appendix C), encouraging the model to explore novel perspectives while referencing existing successful paths.

To maintain constant computational throughput, we allocate a replenishment budget $N_{\text{dse}} = N - |\mathcal{K}_t|$. Exploration is triggered when the kernel size $|\mathcal{K}_t|$ exceeds a density threshold δ relative to either the total width N (for math tasks) or the number of available options N_{option} (for finite-choice tasks). Upon activation, the module executes N_{dse} sampling trials targeting unexplored clusters or low-confidence branches.

The candidate pool is updated via a competitive expansion mechanism: new trajectories τ_{new} with novel answers initialize new groups, while those matching an existing group G_k replace the current representative $\hat{\tau}_k$ only if they yield higher confidence ($S(\tau_{\text{new}}) > S(\hat{\tau}_k)$). This ensures that the representative set $\mathcal{R}_t$ remains both diverse and high-quality.

3.3 Multi-Stage Aggregation: The Convergence Engine

To address the inefficiency of one-shot sampling and leverage the LLM's superior discriminative capability over generative output, MSA redistributes the inference budget into sequential refinement rounds. Instead of unguided generation, MSA reformulates candidate selection as a Multiple-Choice Question (MCQ) task.

In each iteration t, we construct an MCQ prompt $x_t = \mathcal{T}_{\text{MCQ}}(Q, \text{Options}(\mathcal{R}_t))$ using high-quality representatives $\mathcal{R}_t$ from previous steps (see Appendix C). This constrains the search space, allowing the model to discriminate among optimized candidates. We then execute a batch of N sampling trials conditioned on x_t. These trajectories are subsequently re-injected into the CKS module, effectively closing the synergistic loop (CKS $\rightarrow$ DSE $\rightarrow$ MSA) and creating a virtuous cycle of refinement; as the loop iterates, the framework systematically discards implausible paths and reinforces the most promising reasoning chains, thereby progressively purifying the solution space.

The process terminates under two conditions: (1) Convergence: the candidate set collapses to a unique consensus ($|\text{Options}(\mathcal{R}_t)| = 1$), which is then output as the final answer; (2) Budget Exhaustion: the maximum iteration $T_{\max}$ is reached, and the trajectory with the highest confidence from the final round is selected. This recursive density concentration ensures the system converges on the most robust reasoning path under a constrained total budget.

4 Experiments

4.1 Experimental Settings

4.1.1 Datasets and Models

We evaluate our framework on four challenging reasoning benchmarks: AIME 2024(Zhang and Math-AI, 2024), AIME 2025(Zhang and Math-AI, 2025), AMC 23(Zhang and Math-AI, 2023), and GPQA-Diamond(Rein et al., 2023). To verify scalability across model sizes, we employ the Qwen3(Yang et al., 2025) family, specifically the 1.7B and 4B parameter variants, as our backbone LLMs.(See Appendix A for details.)

4.1.2 Baselines

To rigorously evaluate the effectiveness of ConMA, we benchmark it against a comprehensive suite of Test-Time Scaling (TTS) strategies. We utilize Skywork-Reward-V2-Llama-3.1-8B-40M(Liu et al., 2025a) as the unified reward model for all verifier-guided approaches. The baselines are categorized into three distinct groups(See Appendix B for details.):

Independent Sampling Baselines: We evaluate independent sampling on Qwen3-1.7B(Yang et al., 2025), Qwen3-4B(Yang et al., 2025), Qwen3-32B(Yang et al., 2025), and QwQ-32B(Team, 2025) with Greedy Decoding method.

Trajectory-level methods: This category treats the entire trajectory as an atomic unit for selection. We compare against: (1) Self-Consistency (SC-MV)(Wang et al., 2023); (2) Outcome Reward Model(ORM) based BoN(Brown et al., 2024) : ORM-BoN and Weighted ORM-BoN (ORM-WBoN);(3) Confidence based BoN(Kang et al., 2025) : Conf-BoN, Conf-WBoN.

Token-level methods: This category involves step-aware search strategies driven by Process Reward Models (PRM). We include Beam Search(Snell et al., 2024) and Diverse Verifier Tree Search (DVTS)(Liu et al., 2025b), which intervene during the decoding process to refine the reasoning path.

4.1.3 Implementation Details

Standard Settings. To ensure fair comparison, we standardize decoding parameters across all methods using temperature $T = 0.7$ and nucleus sampling $p = 0.9$. For baselines, we strictly control the inference budget: for trajectory-level methods (e.g., BoN, SC-MV), we sample exactly $N = 64$ trajectories in a single pass; for token-level search methods, we configure a beam width of $M = 4$ with $N = 64$ candidate samples per step, selecting the highest-scoring path as the final answer.All experiments are repeated 10 times independently, and we report the average performance.

ConMA Configuration. We constrain ConMA to the same $N = 64$ total budget by setting a per-round width $n = 8$ and maximum iterations $T_{\max} = 4$. The total cost is bounded by $T_{\max} \times (n + \max(N_{\text{dse}})) = 64$, ensuring zero budget overshoot.

Hyperparameters. Internal parameters are set as follows: (1) Confidence: $\alpha = 0.3, \beta = 0.7$ with a tail window $W = 128$ to prioritize conclusion certainty. (2) Aggregation: $\lambda = 0.5$ to balance consensus and quality. (3) Kernel Filtering: Nucleus threshold $\rho = 0.75$ to prune low-quality trajectories. (4) DSE Trigger: Density threshold $\delta = 0.5$ to initiate exploration. Further details and sensitivity analyses are provided in Appendix D.

Models & TTS	AIME25	AIME24	AMC23	GPQA-Diamond
Independent Sampling				
QwQ-32B(Team, 2025)	69.0	76.2	96.4	61.2
Qwen3-32B(Yang et al., 2025)	70.4	78.5	98.3	**62.3**
Qwen3-4B(Yang et al., 2025)	63.3	72.4	92.6	52.5
Qwen3-1.7B(Yang et al., 2025)	34.8	46.7	83.2	36.5
TTS methods w. Qwen3-4B				
SC-MV(Wang et al., 2023)	73.3	80.0	97.5	54.0
ORM-BoN(Brown et al., 2024)	73.7	76.7	90.2	57.6
ORM-WBoN(Brown et al., 2024)	73.3	78.7	95.0	58.5
Conf-BoN(Kang et al., 2025)	73.4	77.4	97.9	58.1
Conf-WBoN(Kang et al., 2025)	72.1	78.2	97.5	58.6
Beam Search(Snell et al., 2024)	77.3	81.3	98.5	60.2
DVTS(Liu et al., 2025b)	78.7	82.0	99.0	61.8
ConMA (Ours)	**80.0**	**84.3**	**99.5**	61.6
TTS methods w. Qwen3-1.7B				
SC-MV(Wang et al., 2023)	46.7	66.7	92.5	38.4
ORM-BoN(Brown et al., 2024)	46.7	69.8	80.0	44.4
ORM-WBoN(Brown et al., 2024)	50.0	70.0	80.0	45.1
Conf-BoN(Kang et al., 2025)	49.0	69.7	92.8	45.2
Conf-WBoN(Kang et al., 2025)	49.3	70.3	93.0	44.8
Beam Search(Snell et al., 2024)	48.3	71.3	92.3	47.3
DVTS(Liu et al., 2025b)	50.7	71.7	93.6	48.4
ConMA (Ours)	51.2	72.1	95.3	48.9

Table 1: **Comparison of accuracy (%) on four benchmarks.** The table reports results for Independent Sampling (top) and various TTS methods with a budget of $N = 64$ applied to Qwen3-4B (middle) and Qwen3-1.7B (bottom). **Bold** indicates the best overall performance, while underlined values mark the best within each group.

4.2 Results

4.2.1 Main Results: Superiority and Scaling Efficiency

Table 1 summarizes the performance across all benchmarks. ConMA consistently outperforms strong search-based and verifier-guided baselines without requiring external reward models.

State-of-the-Art Performance. On Qwen3-4B, ConMA achieves dominant results, reaching 80.0% on AIME 2025 and 84.3% on AIME 2024, surpassing the strongest baseline (DVTS) by 1.3% and 2.3%, respectively. This advantage extends to GPQA-Diamond (61.6%) and the smaller Qwen3-1.7B on AMC 23 (95.3%), confirming that ConMA's outcome-oriented evolution is more effective for complex reasoning than fine-grained token-level search.

Surpassing Larger Models. Notably, ConMA enables small models to outperform significantly larger baselines. On AIME 2025, Qwen3-4B with ConMA (80.0%) exceeds both Qwen3-32B (70.4%) and QwQ-32B (69.0%) by approximately 10%. Similarly, Qwen3-1.7B with ConMA achieves 72.1% on AIME 2024, effectively matching the standard performance of the 4B base model (72.4%).

4.2.2 Compute Efficiency: Accelerating Convergence

We assess efficiency by analyzing scaling behavior and resource consumption in Figures 3 and 4. ConMA demonstrates a significantly steeper scaling trajectory than baselines, notably matching the peak performance of standard methods ($N = 64$) on AIME 2025 (using Qwen3-4B) with an average of only 18 samples. This adaptive saturation breaks the linear cost dependency of fixed-budget baselines, resulting in a $> 70\%$ **reduction** in total token generation while effectively identifying optimal solutions without exhaustive sampling.

5 Analysis

In this section, we delve into the underlying mechanisms that drive the performance gains of ConMA,

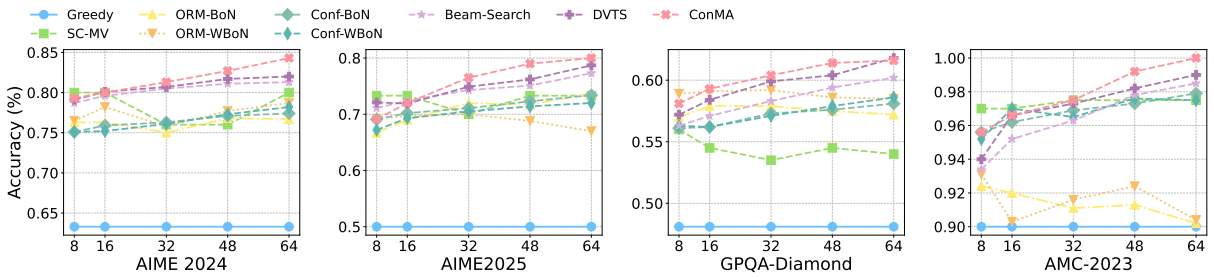

Figure 3: **Performance scaling across sampling budgets** ($N = 8 \ldots 64$)**.** ConMA consistently demonstrates superior accuracy and scaling efficiency on four benchmarks using Qwen3-4B. Note that N denotes the *maximum* allowance for ConMA (due to adaptive stopping), whereas it represents the fixed sample count for baselines.

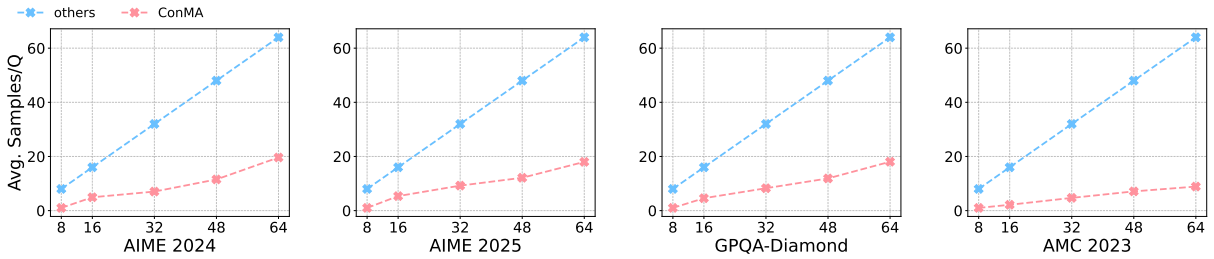

Figure 4: **Resource consumption on Qwen3-4B: Allocated vs. Actual.** Unlike fixed-budget baselines where cost grows linearly ($y = x$), ConMA utilizes adaptive early-stopping to saturate actual usage at low levels. It consistently consumes only a fraction of the maximum budget N, ensuring computational efficiency.

focusing on its effectiveness in evolving the solution space and its efficiency in resource allocation.

5.1 Efficiency Analysis: Candidate-Set Evolution

ConMA's strength lies in its ability to iteratively refine the candidate pool, overcoming the static constraints of one-shot sampling.

Beyond Simple Majority: The Recovery Capability. Traditional Self-Consistency is limited by the initial sampling distribution; if the correct answer is not the majority, SC typically fails. ConMA treats initial samples as a "seed" rather than a final result. By pruning low-confidence noise via CKS, the framework concentrates on a refined kernel of survivors. We observe a robust recovery capability: even when the correct answer is initially in the minority or "hidden" in the tail, the iterative MSA promotes these traces by leveraging the model's discriminative power. This confirms that intrinsic confidence is a more reliable signal for reasoning than raw frequency.

Synergy of Exploration and Pruning. The effectiveness is amplified by the interplay between DSE and the feedback loop. Rather than independent generation, ConMA uses kernel representatives to anchor subsequent exploration, significantly expanding the search space into semantically distinct logical branches. By conditioning new trials on high-quality survivors, the framework leverages intermediate reasoning paths instead of treating samples as isolated events. This progressive refinement mitigates sampling bias in smaller models, enabling them to synthesize complex solutions through iterative accumulation.

5.2 Adaptive Computation: Efficiency via Difficulty Stratification

ConMA adaptively modulates computational intensity based on task difficulty, achieving a superior Pareto frontier between latency and accuracy. We visualize this behavior in Figure 5, showing the distribution of convergence rounds across AIME datasets.

The results reveal a clear bimodal stratification of reasoning effort. For straightforward questions—comprising approximately 86.7% of AIME 2025 (Stage 1-2)—ConMA triggers early convergence. Crucially, this early-stopping is highly reliable: in Stage 1 of AIME 2025, the system achieved a 100% (15/15) success rate, and 83.3% (10/12) in AIME 2024. By identifying a dominant "Kernel" early, the framework effectively prunes redundant search branches, preventing the stochastic

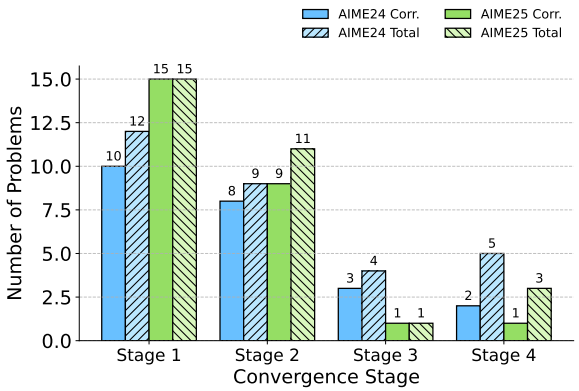

Figure 5: **Distribution of convergence stages and accuracy of Qwen3-4B.** The bar chart illustrates the number of problems that reached consensus at each stage ($T = 1$ to 4) on AIME 2024 and AIME 2025. Notably, ConMA achieves perfect accuracy ($15/15$) in Stage 1 of AIME 2025, demonstrating highly reliable early-stopping. As complexity increases, the framework adaptively allocates more stages for deep reasoning.

noise accumulation inherent in static large-batch sampling.

In contrast, for high-complexity questions (Stage 3-4), the system intelligently extends its reasoning depth. The sustained accuracy in these late-stage instances (e.g., 75% in AIME 2024 Stage 3) indicates that the additional budget is utilized for active logical refinement rather than blind repetition. While the maximum budget is $N = 64$, ConMA's dynamic exit strategy reduces the average sample count to 19.62 for AIME 2024 and 18.0 for AIME 2025. This represents an over 70% reduction in inference cost, proving that ConMA intelligently concentrates computation only where semantic ambiguity necessitates deeper exploration. This demonstrates that ConMA concentrates its "thinking time" only where semantic ambiguity necessitates deeper exploration.

5.3 Ablation Study

To assess the individual contributions of ConMA's core modules, we conducted an ablation study on the AIME 2025 benchmark using Qwen3-4B. Crucially, to ensure a fair comparison, all ablation variants were evaluated under the same strict maximum sampling budget of $N = 64$. We systematically removed key components: (1) w/o CKS, where confidence-based pruning is replaced by random selection; (2) w/o AE, which excludes adversarial path generation; and (3) w/o MSA, which restricts the framework to a static, one-pass evaluation (effectively utilizing the full budget $N = 64$ in a single parallel generation step).

Variant	Configuration	Accuracy	Δ
ConMA (Full)	Complete Framework	**80.0**	–
w/o CKS	Random Kernel Selection	72.1	-7.9
w/o AE	No Adversarial Exploration	75.2	-4.8
w/o MSA	One-Pass Static Selection	73.4	-6.6

Table 2: **Ablation results on AIME 2025 using Qwen3-4B.** Accuracy drops (Δ) highlight the contribution of each module under the same compute budget ($N = 64$).

The results in Table 2 demonstrate the necessity of a holistic design. CKS proves to be the most critical component ($\Delta = -7.9\%$), validating that filtering based on intrinsic confidence is essential for suppressing noise even when sample size is large. MSA follows closely ($\Delta = -6.6\%$), confirming that iterative refinement significantly outperforms static ranking (Best-of-N) by allowing the model to self-correct. Finally, excluding AE leads to a 4.8% drop, indicating that diversity injection effectively prevents the system from collapsing into local optima.

6 Conclusion

We presented ConMA, a training-free TTS framework that transforms static inference budgets into an adaptive sample–filter–diversify–select loop. By utilizing the model's intrinsic confidence for pruning and its discriminative strengths for iterative refinement, ConMA enables smaller models to rival significantly larger counterparts.

Our evaluation on challenging mathematical benchmarks confirms that ConMA pushes the accuracy frontier—achieving 80.0% on AIME 2025 with a 4B model—while reducing the average inference cost by over 70%. This efficiency stems from a dynamic early-stopping mechanism that allocates computation according to task difficulty. Ultimately, ConMA demonstrates that the effectiveness of test-time scaling depends not merely on the quantity of samples, but on the principled management of candidate evolution, providing a robust and resource-efficient blueprint for scaling LLM reasoning at inference time.

Limitations

Despite its strong performance, ConMA faces two primary limitations. First, its efficacy is tied to the base model's self-calibration and discriminative

maturity. Since CKS relies on intrinsic probabilities, a model that is overconfident in erroneous paths may prematurely prune correct solutions, while insufficient discrimination during the MSA stage risks amplifying a "consensus of errors". Second, the framework is currently optimized for objective reasoning tasks with verifiable answers. Extending ConMA to open-ended domains remains non-trivial, as the absence of unique gold-standard answers complicates the answer-grouping and discriminative MCQ formulation required for effective refinement.

Ethics Statement

This work complies with the ACL Ethics Policy. We do not foresee any direct negative social impacts or ethical concerns resulting from this work.

References

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901.

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

Colin Hong, Xu Guo, Anand Chaanan Singh, Esha Choukse, and Dmitrii Ustiugov. 2025. Slim-SC: Thought pruning for efficient scaling with self-consistency. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34500–34517, Suzhou, China. Association for Computational Linguistics.

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

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

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

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

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. 2025a. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *Preprint*, arXiv:2507.01352.

Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. 2025b. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *Preprint*, arXiv:2502.06703.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. Gpqa: A graduate-level google-proof qa benchmark. *Preprint*, arXiv:2311.12022.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

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

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. *International Conference on Learning Representations (ICLR)*.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Jonathan Uesato, Nate Kushman, Ramana Kumar, H. Francis Song, Noah Yamamoto Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2023. Solving math word problems with process-based and outcome-based feedback.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Deli Li, Yunjie anduz, Zhifang Wu, and W Liu. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *ACL 2024*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *International Conference on Learning Representations (ICLR)*.

Zeng Xingshan Liu Weiwen Wang Yufei Li Liangyou Wang Yasheng Shang Lifeng Jiang Xin Liu Qun Wong Kam-Fai Wang, Zezhong. 2025. Stepwise reasoning checkpoint analysis: A test time scaling method to enhance llms' reasoning. *EMNLP 2025*.

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

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 36.

Yuandong Tian Jiawei Zhao Yichao Fu, Xuewei Wang. 2026. Deep think with confidence. *arxiv*.

Yifan Zhang and Team Math-AI. 2023. American mathematics competition 2023.

Yifan Zhang and Team Math-AI. 2024. American invitational mathematics examination (aime) 2024.

Yifan Zhang and Team Math-AI. 2025. American invitational mathematics examination (aime) 2025.

Jiahang Xu Li Lyna Zhang Fan Yang Mao Yang Zhenting Qi, Mingyuan MA. 2025. Mutual reasoning makes smaller llms stronger problem-solver. *ICLR 2025 Poster*.

A Dataset Configuration and Metrics

In this section, we provide detailed statistics and evaluation protocols for the benchmarks used in our experiments.

A.1 Benchmark Statistics

We evaluate our framework on four diverse and challenging benchmarks designed to assess mathematical reasoning and expert-level knowledge. The details are as follows:

- **AIME 2024 & AIME 2025**: These datasets consist of problems from the American Invitational Mathematics Examination (AIME) held in 2024 and 2025. AIME problems serve as a rigorous test bed for advanced mathematical reasoning, requiring models to perform multi-step logical deductions and handle complex arithmetic to reach integer solutions. They represent the frontier of difficulty for current LLMs in competition mathematics.

- **AMC 2023**: This dataset includes problems from the 2023 American Mathematics Competitions (specifically AMC 10 and AMC 12). It represents an intermediate level of difficulty, bridging the gap between foundational arithmetic and Olympiad-level challenges. We use this dataset to evaluate the model's robustness and accuracy on pre-Olympiad standard problems.

- **GPQA-Diamond** (Rein et al., 2023): This is the most challenging subset of the Google-Proof Q&A (GPQA) benchmark, featuring high-difficulty graduate-level questions across biology, physics, and chemistry. It is designed to assess the model's ability to reason over specialized expert knowledge where mere information retrieval is insufficient. We utilize the Diamond subset to verify the generalization of our method beyond pure mathematics.

A.2 Evaluation Metrics

To rigorously assess the performance of ConMA, we employ the following metrics and extraction protocols:

Answer Extraction. Since our models generate free-form Chain-of-Thought (CoT) reasoning, we employ a strict **rule-based extraction script**. For mathematical datasets (AIME, AMC), we extract the final numerical answer contained within the LaTeX box format (i.e., $\backslash\text{boxed}\{answer\}$). For multiple-choice tasks (GPQA), we extract the final selected option character. If the extraction fails or the format is invalid, the sample is marked as incorrect.

Accuracy (Pass@1). Accuracy measures the proportion of questions for which the model's final aggregated answer exactly matches the ground truth. In the context of our ConMA framework, this reflects the performance of the converged solution after the multi-stage aggregation process.

Pass@K. Pass@K quantifies the theoretical potential of the model. It represents the probability that at least one correct solution exists within K independent generations. Formally, for a budget of K samples where c is the number of correct samples, it is calculated as:

$$\text{Pass@}K = 1 - \frac{\binom{N-c}{K}}{\binom{N}{K}} \tag{6}$$

where N is the total pool size used for estimation. In our analysis, we use Pass@K to demonstrate the gap between the model's generative capability (potential) and its selection capability (realized accuracy).

B Baseline Implementations

We detail the configuration and hyperparameters for all baseline methods compared in the main paper. To ensure a fair comparison, all sampling-based methods are strictly constrained to the same total sampling budget of $N = 64$ trajectories. For all verifier-guided approaches (ORM, Beam Search, DVTS), we utilize **Skywork-Reward-V2-Llama-3.1-8B-40M**(Liu et al., 2025a) as the unified reward model.

B.1 Independent Sampling Baselines

To establish performance bounds across different model scales, we evaluate the **Qwen3** family and **QwQ** models using standard decoding strategies:

- **Greedy Decoding:** We utilize Greedy Search (temperature $\tau = 0$) on **Qwen3-1.7B**, **Qwen3-4B**, **Qwen3-32B**(Yang et al., 2025), and **QwQ-32B**(Team, 2025). This serves as the deterministic lower bound for performance.

B.2 Trajectory-level TTS (Best-of-N Variants)

These methods generate $N = 64$ independent trajectories first (using $\tau = 0.7$, top-$p = 0.9$) and then apply different selection strategies to determine the final answer.

Self-Consistency (SC-MV) (Wang et al., 2023) This method applies Majority Voting to the N generated answers. The answer with the highest frequency of occurrence in the final answer set is selected. No external reward model is involved.

ORM-based Methods (Brown et al., 2024) These methods leverage the Skywork-Reward model as an Outcome Reward Model (ORM) to score the final complete trajectories.

- **ORM-BoN:** The standard Best-of-N approach. It selects the single trajectory with the highest scalar reward score assigned by the ORM.
- **ORM-WBoN (Weighted Best-of-N):** This variant aggregates scores to improve robustness. It first groups the N trajectories into clusters based on semantic equivalence of their final answers. For each cluster C_k, we calculate a cluster score $S(C_k)$ by summing the ORM scores of all trajectories within that cluster: $S(C_k) = \sum_{\tau \in C_k} \text{ORM}(\tau)$. The answer corresponding to the cluster with the highest total score is selected.

Confidence-based Methods (Yichao Fu, 2026) These methods rely on the LLM's intrinsic confidence (calculated via the average log-probability of tokens in the generated solution) rather than an external reward model.

- **Conf-BoN:** Selects the single trajectory with the highest average log-probability.
- **Conf-WBoN:** Follows the same logic as ORM-WBoN but uses confidence scores. Trajectories are clustered by answer, and the final selection is based on the cluster with the highest sum of confidence scores.

B.3 Token-level TTS (Search-based Methods)

Unlike trajectory-level methods, these approaches intervene during the decoding process. We utilize the Skywork model as a Process Reward Model (PRM) to score intermediate steps.

Beam Search (Snell et al., 2024) We implement a PRM-guided Beam Search. At each reasoning step, the method maintains a set of active beams. It expands candidates, scores them using the PRM, and retains the top-M high-scoring partial paths for the next step. To align with the $N = 64$ budget, we set the beam width and expansion factor such that the total number of candidate evaluations approximates the cost of generating 64 full trajectories.

Diverse Verifier Tree Search (DVTS) (Liu et al., 2025b) DVTS extends Beam Search by explicitly promoting diversity. It initializes M independent subtrees. In each decoding step, instead of a global top-k selection, DVTS enforces a "subtree isolation" strategy: it samples N/M candidates within each subtree and retains the best path for that specific subtree. We configure DVTS with $M = 4$ subtrees and a per-step sampling width of 16 (totaling $4 \times 16 = 64$ candidates per step) to maintain a computational budget comparable to the $N = 64$ baseline while preventing mode collapse.

C Prompt Templates

To facilitate reproducibility, we present the exact prompt templates used in the ConMA framework. We categorize the prompts into **Mathematical Reasoning** and **Multiple-Choice QA**. Note that for QA tasks, it is critical to retain the **Original Options** (A, B, C, D) in the context to ensure the model selects a valid final answer.

C.1 Standard Generation Prompts

For the initial generation phase, we use standard Chain-of-Thought (CoT) instructions.

Initial Prompt: Mathematical Reasoning

Please solve the following math problem.
Question: *{question}*
Please think step by step and output the final answer in the format \boxed{X}.

Initial Prompt: Multiple-Choice QA

Please solve the following multiple-choice question.
Question: *{question}*
Options: *{options_str}*
Please think step by step and output the final answer in the format \boxed{X}.

C.2 Diversity-Seeking Exploration (DSE) Prompts

In the DSE phase, the goal is to explore diverse possibilities. For **Math**, the generated representatives *become* the options. For **QA**, the representatives serve as *analyses* to help choose the correct *Original Option*.

DSE Prompt: Mathematical Reasoning

Please solve the following multiple-choice question. Question: *{question}*
Review the **Candidate Options** below. They are ordered based on preliminary analysis, but a counter-intuitive possibility may be hidden within. Select the strictly correct option based on logic and calculation.
Candidate Options: *{choices_text}*
Please verify each option step-by-step and select the most correct one. Provide detailed steps and the final answer in the format \boxed{X}.

DSE Prompt: Multiple-Choice QA

Please solve the following multiple-choice question. Question: *{question}*
Original Options: *{options_str}*
Review the **Candidate Analyses** below. They are ordered based on preliminary analysis, but a counter-intuitive possibility may be hidden within. Select the strictly correct Original Option based on logic.
Candidate Analyses: *{choices_text}*
Please verify each analysis step-by-step against the Original Options. Select the strictly correct option and output the final answer in the format \boxed{X}.

C.3 Multi-Stage Aggregation (MSA) Prompts

In the MSA phase, we retain the Original Options for QA to ensuring the "Critical Review" validates the reasoning against the specific choices provided.

MSA Prompt: Mathematical Reasoning

Question: *{question}*
We have generated some preliminary solutions (ranked by likelihood): *{candidates_text}*
Please perform a **Critical Review** (Sanity Check) on the above solutions: 1. Do they interpret the question correctly? 2. Are there any calculation errors in the steps? 3. Is there a logic gap or an edge case ignored?
If the current solutions are robust and correct, please verify them and explain why. If you find a logical flaw, please propose a **corrected** solution path and answer.

MSA Prompt: Multiple-Choice QA

Question: *{question}*
Original Options: *{options_str}*
We have generated some preliminary reasoning paths (ranked by likelihood): *{candidates_text}*
Please perform a **Critical Review** (Sanity Check) on the above paths: 1. Do they interpret the question correctly? 2. Are there any factual errors in the reasoning? 3. Is there a logic gap or an edge case ignored? 4. Do they strictly correspond to one of the Original Options?
If the current paths are robust and correct, please verify them and explain why. If you find a logical flaw, please propose a **corrected** reasoning path and answer.

D Hyperparameter Analysis and Sensitivity

In this section, we present a rigorous component-level evaluation to justify our hyperparameter selection. Instead of computationally expensive end-to-end retraining, we adopted an **Offline Proxy Evaluation** protocol. We utilized the raw sampling data generated during the validation phase on AIME 2024 to construct a fixed, labeled candidate pool. This allows us to isolate and quantify the discriminative capability of each module independent of generative randomness.

D.1 Experimental Setup: The Offline Candidate Pool

To ensure a rigorous and controllable evaluation, we constructed a dataset $\mathcal{D}_{\text{pool}}$ consisting of a representative subset of 10 problems selected from the AIME 2024 training set. These problems were carefully stratified by difficulty (Easy to Very Hard) and mathematical domain (Algebra, Geometry, Number Theory, Combinatorics) to ensure comprehensive coverage of the model's reasoning capabilities.

For each problem x_i, we generated $N = 64$ reasoning trajectories $\{\tau_{i,1}, \ldots, \tau_{i,64}\}$ using the base Qwen3-4B model. We then applied **Automatic Ground-Truth Labeling**: each trajectory was marked as positive ($y_{i,j} = 1$) if its final boxed answer matched the ground truth, and negative ($y_{i,j} = 0$) otherwise. This process resulted in a static testbed of 640 labeled trajectories, which served as the basis for the following component-level sensitivity analyses.

D.2 Confidence Scoring Analysis (α, β, W)

Objective: To optimize the scoring function $S(\tau) = \alpha \cdot \mathcal{C}(\tau_{\text{full}}) + \beta \cdot \mathcal{C}(\tau_{\text{tail}})$ for distinguishing correct reasoning from incorrect ones.

Method: We treated this as a binary classification ranking problem. For each problem in $\mathcal{D}_{\text{pool}}$, we ranked the 64 trajectories based on $S(\tau)$ and calculated the **Area Under the ROC Curve (AUC)** and **Pearson Correlation** (r) with the ground truth labels. We varied β (with $\alpha = 1 - \beta$) and the tail window W.

Result: As detailed in Table 3, a window of $W = 128$ combined with a heavy tail weight $\beta = 0.7$ achieves the highest AUC (0.762). This indicates that while the final derivation is the strongest signal of correctness, completely ignoring the full reasoning context ($\beta = 1.0$) slightly degrades discriminative performance.

Table 3: **Discriminative Power (AUC) of Confidence Scoring.** We evaluate the impact of the tail weight (β) and window size (W) on ranking accuracy. The results indicate a clear "sweet spot" at $W = 128$ and $\beta = 0.7$, suggesting that while the conclusion (τ_{tail}) is the primary signal, incorporating the full reasoning context (τ_{full}) is essential for robustness.

Window W	**Tail Weight** β				
	0.0	0.3	0.5	**0.7**	1.0
64 tokens	0.682	0.710	0.735	0.748	0.730
128 tokens	0.685	0.715	0.741	**0.762**	0.744
256 tokens	0.684	0.712	0.738	0.755	0.740

D.3 Group Aggregation Analysis (λ)

Objective: To determine the optimal weight λ for merging "Vote Count" (Consensus) and "Average Confidence" (Quality) into the Group Score $S(G_k)$.

Method: Using the same offline pool, we merged trajectories into semantic groups and evaluated the **Top-1 Selection Accuracy** of the resulting group scores under varying λ.

Table 4: **Impact of Group Weight (λ) on Selection Accuracy.** We compare pure quality-based ranking ($\lambda = 0$), pure voting ($\lambda = 1$), and hybrid approaches. The hybrid configuration ($\lambda = 0.5$) significantly outperforms standard majority voting (70% vs. 64%), validating that intrinsic model confidence can correct popular but erroneous consensus.

Method	$\lambda = 0.0$	$\lambda = 0.25$	$\lambda = \mathbf{0.50}$	$\lambda = 0.75$	$\lambda = 1.0$
Focus	*Quality*	-	***Hybrid***	-	*Vote*
Acc (%)	62.0	66.0	**70.0**	68.0	64.0

Result: Table 4 validates the hybrid strategy. Relying solely on voting ($\lambda = 1.0$) yields 64% accuracy (equivalent to standard Majority Voting). By incorporating model confidence ($\lambda = 0.5$), we improve selection accuracy to 70%, proving that low-frequency but high-confidence solutions are effectively rescued by our scoring mechanism.

D.4 Kernel Pruning Analysis (ρ)

Objective: To evaluate the trade-off between noise reduction and recall preservation in the CKS module.

Method: We simulated the Nucleus Sampling process on the offline groups. We measured two

metrics: (1) **Noise Reduction Ratio (NRR)** and (2) **Recall Retention (RR)**.

Table 5: **Pruning Efficiency (Noise Reduction vs. Recall).** We evaluate different cumulative probability thresholds ρ. A threshold of $\rho = 0.75$ is selected as the optimal operating point: it effectively filters out nearly half of the incorrect groups (48.5% NRR) while retaining the correct answer in the kernel with near-certainty (98.1% RR).

Threshold ρ	**Recall (RR) ↑**	**Noise Red. (NRR) ↑**	**Verdict**
0.50 (Aggressive)	84.5%	$\mathbf{75.0}\%$	Unsafe
0.60	91.2%	62.4%	Suboptimal
0.75 (Default)	$\mathbf{98.1}\%$	48.5%	**Optimal**
0.90 (Conservative)	99.5%	15.2%	Inefficient

Result: We selected $\rho = 0.75$ as it maintains a near-perfect recall (98.1%) while filtering out nearly half (48.5%) of the incorrect groups. Lower thresholds risk discarding the truth, while higher thresholds fail to reduce the search space for the subsequent exploration phase.