PiCSAR: Probabilistic Confidence Selection and Ranking for Reasoning Chains

Joshua Ong Jun Leang[1,2] **Zheng Zhao**[2] **Aryo Pradipta Gema**[2] **Sohee Yang**[3]
Wai-Chung Kwan[2] **Xuanli He**[3] **Wenda Li**[2] **Pasquale Minervini**[2,4]
Eleonora Giunchiglia[1] **Shay B. Cohen**[2]
[1]Imperial College London [2]University of Edinburgh [3]UCL [4]Miniml.AI
{j.ong25,e.giunchiglia}@imperial.ac.uk scohen@inf.ed.ac.uk

Abstract

Best-of-n sampling improves the accuracy of large language models (LLMs) and large reasoning models (LRMs) by generating multiple candidate solutions and selecting the one with the highest reward. A key challenge for reasoning tasks is designing a scoring function that can identify correct reasoning chains without access to ground-truth answers. We propose **P**roba**bi**listic **C**onfidence **S**election **A**nd **R**anking (PiCSAR): a simple, training-free method that scores each candidate generation using the joint log-likelihood of the reasoning and final answer. This method uses both the scores of the reasoning path (*reasoning confidence*) and the final answer (*answer confidence*). PiCSAR achieves substantial gains across several benchmarks (+11.7 on AIME2024, +9.81 on AIME2025), outperforming baselines with at least 2x fewer samples in 20 out of 25 comparisons. Our analysis reveals that correct reasoning chains exhibit higher reasoning and answer confidence levels, justifying the effectiveness of PiCSAR[1].

1 Introduction

Recent studies have shown that LLMs achieve strong performance on complex reasoning tasks (Grattafiori et al., 2024; Team et al., 2024; Hurst et al., 2024). Techniques such as Chain-of-Thought (CoT; Wei et al., 2022; Kojima et al., 2022) aim to enhance the reasoning process by generating explicit intermediate reasoning steps. Building on these advances, large reasoning models (LRMs), LLMs that receive intensive reasoning-focused post-training, such as DeepSeek-R1 (Guo et al., 2025) and Qwen3 (Yang et al., 2025a), solve complex problems by generating long CoT reasoning traces. These traces are often extended via test-time scaling (Muennighoff et al., 2025) and can include reflective self-checking (Yang et al., 2025b).

[1]Code: https://github.com/joshuaongg21/PiCSAR

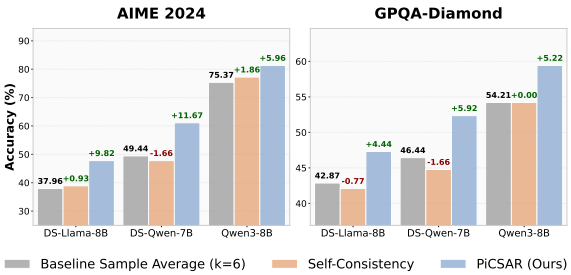


Figure 1: Performance of PiCSAR and Self-Consistency on two reasoning datasets and three models.

Despite these advances, classic decoding approaches such as greedy decoding often fall short of state-of-the-art performance on complex benchmarks (Team et al., 2025; Balunović et al., 2025), emphasising the need for more sophisticated inference-time strategies. *Best-of-N* (BoN) sampling (Stiennon et al., 2020) emerged as an important technique, where n candidate responses are generated, and the highest-scoring one is selected via a reward model (Mudgal et al., 2024; Huang et al., 2025). However, training external reward models can be computationally expensive (Wang et al., 2023a) and vulnerable to distribution shifts (Eisenstein et al., 2023).

This led to the adoption of simpler, training-free BoN variants, such as Self-Consistency (Wang et al., 2023b), which selects the most frequent answer among multiple generated outputs. However, a key limitation of Self-Consistency is its exclusive reliance on the final answer while ignoring the reasoning that leads to it. Extensions such as Universal Self-Consistency (USC; Chen et al., 2023b) prompt the model to identify the most consistent response from a set of candidates. However, USC focuses on majority agreement over full responses, overlooking reasoning-level signals critical to answer quality, such as coherence and plausibility. USC is further constrained by context-window size and reasoning ability (Chen et al., 2023b), proving particularly ineffective with smaller models (Kang et al.,

Findings of the Association for Computational Linguistics: ACL 2026, pages 31511–31544
July 2-7, 2026

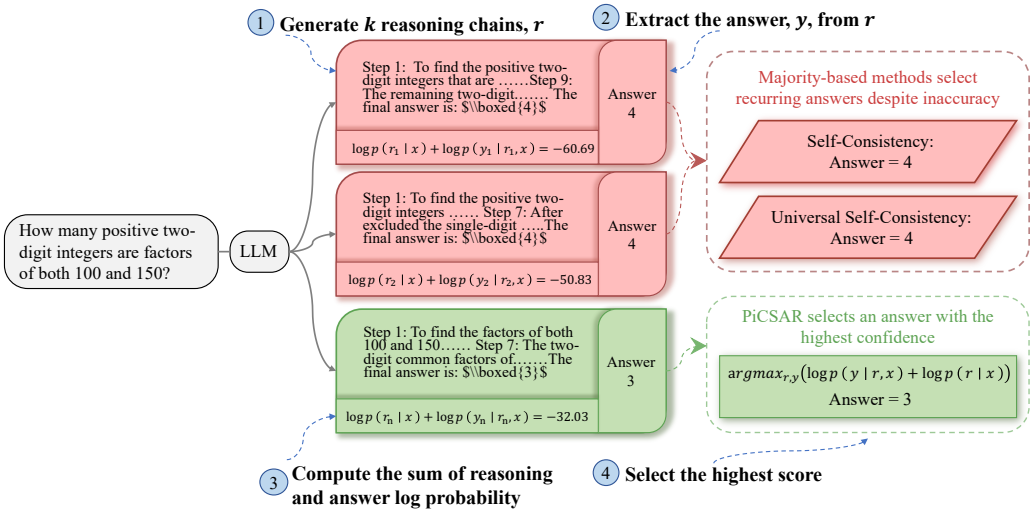


Figure 2: Example with *Llama-3.1-8B* on *MATH500*, where PiCSAR selects the most likely reasoning trace r and answer y by jointly maximising their log-likelihoods $\log p(r \mid x)$ and $\log p(y \mid r, x)$.

2025). Attempts to overcome this by prompting the model to self-evaluate are often ineffective, as explicit confidence is often poorly calibrated (Miao et al., 2024; Taubenfeld et al., 2025).

To address these challenges, we introduce Probabilistic Confidence Selection And Ranking (PiCSAR), a probabilistic confidence method for selecting a reasoning chain r together with its corresponding answer y without requiring any additional training or fine-tuning. Our approach is straightforward to implement and can be used with any LLM or LRM as an inference-time tool. It is based on a new scoring function that, given a prompt x, selects a reasoning chain r and the answer y by maximising their joint conditional likelihood $\log p(y, r \mid x)$. This objective naturally separates into two complementary components. The *reasoning confidence* term $\log p(r \mid x)$ promotes high-probability reasoning sequences by implicitly evaluating the likelihood of the chain given the prompt. The *answer confidence* term $\log p(y \mid r, x)$ quantifies the model's certainty in its final prediction, conditioned on the generated reasoning chain. Figure 2 shows a high-level outline of PiCSAR, and how it can solve instances that Self-Consistency and USC cannot solve correctly.

We evaluate PiCSAR on reasoning tasks across five LLMs and three LRMs, outperforming Self-Consistency and USC in most cases. PiCSAR achieves these gains with far fewer samples, often requiring only $k = 6$ samples to beat baselines using $k = 16$ or 32 samples. PiCSAR substantially improves LRM performance, with Deepseek-R1-distilled-Llama-3 gaining *+13.33% and +7.58%* over Self-Consistency on AIME2024 and GPQA-Diamond, respectively (Figure 1). Unlike USC, which is bounded by the model's reasoning abilities, PiCSAR decouples confidence estimation, allowing smaller models to effectively capture stable reasoning process properties rather than model artefacts (§5.3).

Beyond empirical results, we provide a comprehensive analysis of LLM confidence behaviour. At finer granularity, we analyse answer confidence at the sentence level using *information density*, defined as the ratio of peak-confidence instances to sentence count (peak-to-sentence ratio), which measures how frequently a reasoning chain attains high confidence relative to its length. We find that higher accuracy correlates with high information density within model families (§5.1). In addition, we show that answer confidence positively correlates with downstream accuracy (§5.2).

2 A Joint Probabilistic Method for Reasoning Chain Selection

We propose a training-free method for selecting a reasoning chain from a set of candidates, grounded in a probabilistic framework that leverages the model's confidence as its scoring signal. We frame the selection problem as an approximation of maximum a posteriori (MAP) decoding over the joint space of reasoning chains and final answers.

2.1 Scoring Function and Log-likelihood Decomposition

We denote by $\mathcal{X}$ a set of possible prompts, $\mathcal{R}$ a set of reasoning chains, and $\mathcal{Y}$ the set of possible final answers. For a given input prompt $x \in \mathcal{X}$, our goal is to find the high-confidence reasoning chain $r \in$

$\mathcal{R}$ and its corresponding answer $y \in \mathcal{Y}$. Consider a selection criterion that aims to identify the pair (r, y) with the highest joint conditional probability, $p(r, y \mid x)$. By the chain rule of probability, this decomposes into two distinct components:

$$p(r, y \mid x) = p(y \mid r, x) \cdot p(r \mid x). \quad (1)$$

In log-space, the joint probability becomes the sum of two log-likelihood terms as follows:

$$\text{Score}(r, y) = \underbrace{\log p(r \mid x)}_{\text{Reasoning Confidence}} + \underbrace{\log p(y \mid r, x)}_{\text{Answer Confidence}}. \quad (2)$$

These two terms provide complementary signals regarding the quality of a candidate generation:

- **Reasoning Confidence ($\log p(r \mid x)$):** This term quantifies the model's confidence in generating r given the prompt x. It quantifies the plausibility of the reasoning path itself.
- **Answer Confidence ($\log p(y \mid r, x)$):** measures the model's certainty in the answer y, *conditioned on the reasoning chain it has produced.*

2.2 Probabilistic Confidence Selection And Ranking (PiCSAR)

Directly selecting $r \in \mathcal{R}$, $y \in Y$, where the joint log likelihood $\text{Score}(r, y)$ is maximised over the space of possible pairs, is intractable. We therefore approximate this optimisation with our PiCSAR sampling-based approach, as outlined in Algorithm 1. We first generate k candidate reasoning chains $\{r_1, r_2, \ldots, r_k\}$ from the model's posterior $p(r \mid x)$. Each chain r_i implies a corresponding final answer y_i. We then re-rank these candidates using the PiCSAR scoring function.

The *reasoning confidence* term is obtained by summing the token-level log-probabilities from the model during the generation of r_i. By not applying length normalisation, this term naturally favours more concise and direct reasoning paths as it involves a cumulative sum of individual token log-probabilities. We also consider the length-normalised variant, PiCSAR-N, which focuses more on the impact of log probability per token rather than favouring concise reasoning paths, leading to similar results (details in Appendix C.3).

The *answer confidence* term, $\log p(y \mid r, x)$, however, presents a practical challenge. As the model's distribution is over all possible text continuations, the probability of a final answer is confounded by the likelihood of whatever text might follow it. This makes the raw log-probabilities of different answers fundamentally incomparable. To address this and ensure we can reliably extract a final answer for answer confidence computation, we condition the model on an explicit instruction prompt, denoted as $\langle a \rangle$, which is appended after the reasoning chain. This prompt explicitly asks the model to provide the final answer based on the preceding context (*i.e.,* "*When you see a potential reasoning followed by* $\langle$sep$\rangle$*, output the final answer.*"), with details of the prompt provided in Appendix B. While we extract the answer y directly from the reasoning chain r, we use this augmented prompt to compute the answer confidence.

Our modified objective is thus:

$$\arg\max_{r,y} \left[\log p(r \mid x) + \log p(y \mid \langle a \rangle, r, x)\right]. \quad (3)$$

Methodological Departure from Standard MAP Decoding. While the decomposition in Equation (3) relies on the foundational chain rule, PiCSAR fundamentally differs from standard Maximum A Posteriori (MAP) decoding or beam search. In standard continuous decoding, the joint probability of a CoT sequence is disproportionately dominated by the arbitrary length and local perplexity of the reasoning steps, effectively drowning out the signal of the final deductive answer. This limitation has historically driven the field away from likelihood-based scoring for reasoning tasks, favouring majority voting or externally trained re-

Algorithm 1 Probabilistic Confidence Selection And Ranking (PiCSAR)

Input: Prompt x, number of samples k, instruction prompt $\langle a \rangle$.
Output: Reasoning chain r^* and answer y^*.
Generate Candidates: Independently sample k reasoning chains $\{r_1, r_2, \ldots, r_k\}$ from the model, where each $r_i \sim p(r|x), i = 1, \ldots, k$.
Score Candidates:
for each $i \in \{1, \ldots, k\}$ **do**
  Extract Reasoning Confidence: Retrieve $C_{\text{reason}}(i) = \log p(r_i \mid x)$ from r_i.
  Extract Answer: Extract answer, y_i, from reasoning chain, r_i.
  Compute Answer Confidence: $C_{\text{answer}}(i) = \log p(y_i \mid \langle a \rangle, r_i, x)$.
  Compute Final Score: $\text{Score}(r_i, y_i) = C_{\text{reason}}(i) + C_{\text{answer}}(i)$.
end for
Select Best: Identify the highest-scoring candidate: $i^* = \arg\max_i \text{Score}(r_i, y_i)$.
Return: (r_{i^*}, y_{i^*}).

ward models. By introducing the instructional intervention $\langle a \rangle$, PiCSAR breaks this continuous autoregressive evaluation. It explicitly forces the model to evaluate the logical entailment of the answer independently of the generative probability of the preceding text. This isolates the conditional *answer confidence*, turning Equation (2) into Equation (3) and thereby yielding a robust, training-free ranking mechanism.

The final step is to select the candidate pair with the highest score. As illustrated in Figure 2, the two components of our scoring function play complementary roles. The *reasoning confidence* is the sum of log-probabilities for every token in the reasoning chain. Since these log-probabilities are negative, longer sequences tend to accumulate more negative values (i.e., larger magnitude), and can therefore dominate the overall score (see Appendix G). The *answer confidence* in turn serves as a discriminator, often proving decisive when multiple candidate chains exhibit similar reasoning plausibility.

2.3 Confidence Information Plane

To motivate PiCSAR design, we analyse the distribution of model-generated samples on a 2D "Information Plane", with respect to our two confidence terms (Figure 3). We partition the plane into four quadrants using the median value of each axis. $\log p(y \mid r, x) = -10$ is used when the model fails to answer (*i.e.,* when no answer token is generated and the answer-confidence term cannot be computed). We compared this fallback value (-10) with various other values, and the results are in Appendix C.8. For Llama-3.1-8B on the MATH500 dataset, we see that correct answers (green) are concentrated in the upper-right quadrant (Q1), corresponding to high scores on *both* confidence terms.

The quadrant-wise accuracy breakdown is stark: the upper-right quadrant (Q1) achieves 71.7% accuracy, outperforming other quadrants (Q2: 39.0%, Q3: 31.6%, Q4: 62.2%). High reasoning confidence (Q1 and Q4) leads to a higher performance than a high answer confidence (Q2 and Q3). This is reinforced by a statistical t-test that, while both terms are highly significant predictors of correctness, reasoning confidence is a significantly stronger predictor (t-statistics ≈ 9.111) than answer confidence (t-statistics ≈ 4.753). For more details on the statistical tests, see Appendix E.1. Nevertheless, both confidence measures remain essential components for reasoning chain selection.

This principle can be used as a practical filter;

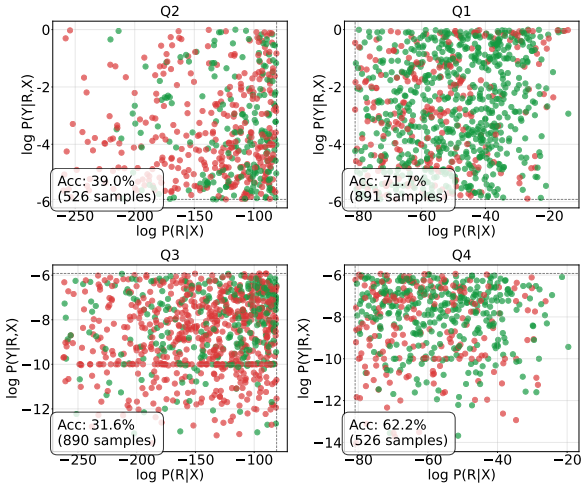


Figure 3: Information plane of MATH500 questions that Llama-3.1-8B predicts correctly and incorrectly ($k = 6$). Quadrants show combinations of answer and reasoning confidence. This pattern is consistent across LLMs, LRMs, and datasets (Appendix E).

tightening the thresholds to the 75th percentile, for instance, isolates a subset of samples with near-perfect accuracy (*i.e.,* 100% on DS-Distilled-Qwen-2.5-7B with AIME2025), providing a mechanism to identify reliable instances (further examples in Appendix E). Overall, our analysis reveals that correct reasoning exhibits higher reasoning and answer confidence, *with reasoning confidence being a substantially stronger predictor of correctness.*

3 Experimental Setup

Models. We evaluate PiCSAR across a diverse set of recent LLMs and LRMs. Our experiments include LLMs from three major families: Llama-3.1-Instruct (8B and 70B; Dubey et al. 2024), Gemma-2-Instruct (9B; Team et al. 2024), and Qwen3 (8B and 32B; Yang et al. 2025a). For the Qwen3 models, we disable the *thinking mode*. For LRMs, we include two distilled models from the DeepSeek-R1 series (DS-distill-Llama-3.1-8B and DS-distill-Qwen-2.5b; Guo et al. 2025), and the Qwen-3-8B model with *thinking mode* enabled. We exclude larger LRMs due to computational cost.

Baselines. We compare against six baselines: *Greedy Decoding* (1); *Self-Consistency* (Wang et al., 2023b) (2); *USC* (Chen et al., 2023b) (3); *p(True)* (Kadavath et al., 2022) (4); *Self-Certainty* (Kang et al., 2025) (5). Confidence-Interval Self-Consistency (CISC; Taubenfeld et al. 2025) is discussed in Appendix C.1, as it involves weight voting. While CISC was originally proposed using *p(True)*, we also report CISC(PiCSAR)

Method	SVAMP		GSM8K		MATH500		GPQA-Diamond		TheoremQA	
	$k=6$	$k=16/32$	$k=6$	$k=16/32$	$k=6$	$k=16/32$	$k=6$	$k=16/32$	$k=6$	$k=16/32$
Gemma-2-9B-Instruct										
Greedy Decoding	87.33		86.64		41.40		29.80		17.14	
Self-Consistency	88.15±0.22	88.89±0.22	87.04±0.24	88.10±0.05	41.60±0.40	43.27±0.23	27.27±0.58	23.91±1.38	15.44±0.12	14.10±0.00
USC	88.63±0.13	-	85.74±0.27	-	42.54±0.37	-	24.33±1.21	-	17.24±0.33	-
p(True)	88.56±0.44	87.89±0.22	88.36±0.22	88.38±0.08	**46.87**±0.07	46.80±0.70	30.30±1.54	33.50±0.17	15.62±0.37	15.98±0.44
Self-Certainty	88.48±0.04	88.33±0.06	87.18±0.08	87.32±0.03	43.93±0.13	43.93±0.08	26.77±0.42	27.41±0.83	14.73±0.28	14.77±0.04
PiCSAR	**89.00**±0.38*	**91.02**±0.59	**88.66**±0.11*	**88.99**±0.20	46.53±0.29*	**47.13**±0.13	**32.32**±0.51*	**34.01**±1.94	**18.62**±0.39*	**18.88**±0.54
Upper Bound	93.44±0.22	95.67±0.38	93.44±0.09	95.60±0.04	58.47±0.27	66.67±0.47	55.22±1.10	82.49±1.02	24.32±0.49	32.40±0.20
Llama-3.1-8B-Instruct										
Greedy Decoding	89.67		87.47		50.40		27.27		17.80	
Self-Consistency	88.33±0.67	89.89±0.11	86.67±0.38	89.52±0.16	46.33±0.13	50.13±0.48	26.09±0.45	26.67±1.34	15.62±0.18	12.72±0.48
USC	89.87±0.23	-	88.22±0.23	-	51.80±1.25	-	25.67±1.54	-	18.88±0.31	-
p(True)	85.33±0.00	83.22±0.91	87.40±0.44	86.59±0.03	47.73±0.66	47.80±0.72	27.27±1.75	26.09±2.07	14.41±0.59	14.10±0.51
Self-Certainty	89.44±0.06	89.49±0.26	87.43±0.24	87.35±0.02	51.04±0.20	51.09±0.16	26.54±0.49	26.30±0.49	14.91±0.13	14.62±0.14
PiCSAR	**91.78**±0.11*	**93.44**±0.89	**89.09**±0.13*	**89.98**±0.23	**53.33**±0.73*	**53.87**±0.70	**29.80**±1.34*	**33.67**±3.06	**20.08**±0.43*	**19.72**±0.39
Upper Bound	96.78±0.11	99.11±0.11	96.15±0.07	98.18±0.04	72.80±0.23	82.20±0.60	65.82±1.50	92.76±0.73	28.20±0.32	37.84±1.13
Qwen3-8B (Non-thinking)										
Greedy Decoding	93.33		92.48		**73.40**		42.23		27.71	
Self-Consistency	92.52±0.33	93.11±0.11	92.29±0.13	91.69±0.11	73.00±0.23	72.27±0.00	47.47±0.29	40.74±1.61	28.33±0.31	28.51±0.33
USC	93.11±0.22	-	**93.24**±0.13	-	73.60±0.12	-	**48.38**±2.06	-	27.88±0.55	-
p(True)	92.44±0.56	91.78±0.44	92.10±0.00	91.22±0.18	72.67±0.24	71.20±0.60	41.25±1.71	36.20±1.44	27.84±0.18	28.28±0.13
Self-Certainty	92.63±0.21	92.83±0.04	92.29±0.07	92.25±0.04	71.94±0.16	71.82±0.14	44.33±0.54	42.29±0.81	27.97±0.66	27.92±0.77
PiCSAR	**93.56**±0.22*	**95.13**±0.22	92.33±0.13*	**93.22**±0.08	**73.67**±0.24*	**73.40**±0.13	46.98±1.01*	**43.69**±1.26	**29.76**±0.58*	**29.17**±0.64
Upper Bound	96.33±0.67	97.89±0.11	95.52±0.00	96.84±0.03	81.13±0.44	83.53±0.24	76.26±1.62	86.36±0.29	34.94±0.00	40.03±0.35
Llama-3.1-70B-Instruct										
Greedy Decoding	**94.33**		93.93		60.20		40.44		**30.79**	
Self-Consistency	92.78±0.56	93.45±0.11	94.00±0.10	93.98±0.13	58.60±0.46	60.80±0.87	42.59±1.02	37.54±0.67	26.55±0.47	25.61±0.00
USC	92.78±0.11	-	93.29±0.20	-	60.60±0.95	-	41.25±1.76	-	27.44±0.67	-
p(True)	93.11±0.78	93.11±0.40	94.51±0.13	94.08±0.23	61.47±1.14	62.33±1.16	41.25±1.61	42.09±2.21	24.45±0.31	24.23±0.61
Self-Certainty	93.02±0.30	93.84±0.01	94.01±0.13	94.04±0.05	61.82±0.08	61.70±0.14	39.84±0.88	38.87±0.67	24.43±0.18	24.56±0.11
PiCSAR	94.10±0.11*	**95.58**±0.22	**94.58**±0.03*	**94.81**±0.13	**63.67**±1.51*	**64.07**±0.87	**46.91**±2.65*	**46.46**±2.59	27.84±0.19*	26.73±0.27
Upper Bound	97.22±0.22	97.78±0.22	96.91±0.03	97.44±0.03	77.07±0.47	81.67±0.18	75.59±0.61	87.71±0.45	40.70±0.20	43.47±0.18
Qwen3-32B (Non-thinking)										
Greedy decoding	92.33		93.24		75.00		**48.48**		29.99	
Self-consistency	92.67±0.33	93.11±0.33	93.62±0.00	93.75±0.08	75.93±0.33	**76.27**±0.12	47.31±1.98	44.44±0.51	30.79±0.00	30.92±0.28
USC	92.44±0.78	-	93.69±0.13	-	76.16±0.64	-	44.90±0.55	-	30.07±0.51	-
p(True)	**93.22**±0.11	93.00±0.69	92.79±0.53	92.91±0.25	74.07±1.07	74.00±0.35	39.90±2.81	38.05±0.94	30.79±0.00	30.08±0.12
Self-certainty	92.63±0.18	92.92±0.16	92.29±0.03	93.45±0.02	71.94±0.09	75.68±0.10	43.07±1.16	43.39±0.73	30.23±0.00	30.61±0.13
PiCSAR	**93.22**±0.22*	**93.55**±0.33	**93.90**±0.28*	**93.88**±0.22	**77.00**±0.18*	75.93±0.13	46.91±1.02*	44.44±2.28	**31.46**±0.04*	**31.42**±0.27
Upper Bound	96.78±0.11	98.00±0.00	96.28±0.13	96.99±0.07	82.27±0.13	83.73±0.07	72.56±1.87	86.20±1.02	39.76±0.00	42.93±0.12

Table 1: **Comparison of model accuracies on LLMs.** $k = \{6, 32\}$ for Gemma-2-9B, Llama-3.1-8B, Qwen3-8B; $k = \{6, 16\}$ for Llama-3.1-70B, Qwen3-32B due to computational constraints. **Bold**: highest, **underline**: equal highest, *: $k = 6$ outperforms $k = 16, 32$ baselines. *PiCSAR with $k = 6$ outperforms larger k in 20/25 cases.*

for a fair comparison. Due to context length limits and computational constraints, we exclude (3), (4), and (5) in LRMs and set $k = 16, 32$ in LLMs.

To isolate each component's contribution in PiCSAR, we include three ablations in Appendix C.2 and C.3: *Reasoning Confidence* ($\max_r(\log p(r \mid x))$), with (6) and without (7) length normalisation, and *Answer Confidence* ($\max_y(\log p(y \mid r, x))$) (8). For LRMs, we compare against (1), (2), (6), (7), (8). We also include the $pass@k$ upper bound, representing the maximum achievable accuracy when at least one of the k candidates is correct. Implementation details can be found in Appendix B.

Datasets. We evaluate LLMs on three maths benchmarks: GSM8K (Cobbe et al., 2021), SVAMP (Patel et al., 2021), MATH500 (Hendrycks et al., 2021), and two scientific reasoning benchmarks GPQA-Diamond (Rein et al., 2024), and TheoremQA (Chen et al., 2023a). We additionally evaluate LRMs on AIME 2024 and 2025, omitted for LLMs due to difficulty. Results are averaged over three runs and reported with standard errors.

4 Experimental Results

Performance on LLMs. In Table 1, we see that when using PiCSAR, Llama models show consistent improvements across all baselines. With $k = 6$ on Llama-3.1-8B, PiCSAR outperforms the best-performing baseline (*i.e.,* Self-Certainty) by 3.26% (26.54% $\rightarrow$ 29.80%) on GPQA-Diamond. On Llama-3.1-70B PiCSAR shows similar gains: 7.07% improvement over Self-Certainty and 5.66% over USC. We observe a similar trend on Gemma-2-9B; at k=6, PiCSAR outperforms Self-Consistency by 4.93%. This outcome aligns with our information-plane analysis (see Figure 3); PiCSAR selects candidates in the top-right, high-accuracy quadrant by maximising the joint score of reasoning and answer confidence. For

Method	SVAMP	GSM8K	MATH500	GPQA-Diamond	TheoremQA	AIME 2024	AIME 2025
DS-Distill-llama-3-8B							
Average	82.11±0.13	73.67±0.32	65.55±0.25	42.87±1.07	26.58±0.06	37.96±1.52	29.63±0.37
Self-Consistency	**86.17**±0.27	74.01±0.70	66.25±0.40	42.10±1.77	27.98±0.87	38.89±1.67	25.00±0.37
PiCSAR	85.67±0.07	**76.42**±0.16	**67.20**±0.60	**47.31**±0.17	**28.02**±0.78	**47.78**±4.01	**33.33**±1.11
Upper Bound	95.67±0.00	92.91±0.35	82.00±0.13	77.27±0.77	36.37±2.83	66.67±5.09	51.11±1.11
DS-Distill-Qwen-2.5-7B							
Average	89.26±0.13	87.29±0.14	72.79±0.16	46.44±1.63	33.11±0.14	49.44±3.06	41.30±1.30
Self-Consistency	90.39±0.20	**89.50**±0.37	73.87±0.25	44.78±1.83	35.88±0.35	47.78±3.40	38.33±3.34
PiCSAR	**91.78**±0.48	88.18±0.07	**74.00**±0.70	**52.36**±2.88	**36.76**±0.44	**61.11**±1.11	**51.11**±1.11
Upper Bound	96.33±0.38	96.79±0.13	83.33±0.18	79.12±2.07	48.59±0.08	72.22±1.11	70.00±0.00
Qwen3-8B							
Average	91.43±0.07	95.43±0.01	80.44±0.10	54.21±0.83	40.83±0.13	75.37±0.19	67.04±2.06
Self-Consistency	91.83±0.33	95.68±0.03	80.40±0.18	54.21±1.68	41.81±0.11	77.23±1.11	65.56±2.58
PiCSAR	**94.33**±0.33	**95.94**±0.04	**80.60**±0.13	**59.43**±1.61	**42.57**±0.27	**81.33**±1.34	**68.89**±2.22
Upper Bound	97.56±0.11	97.54±0.03	84.00±0.12	80.13±0.45	44.71±1.34	87.78±1.11	82.22±1.11

Table 2: **Comparison of model accuracies on LRMs** ($k = 6$). *PiCSAR outperforms the baselines in 19/21 cases.*

the Qwen family, PiCSAR generally leads across benchmarks and sample counts (k). While there are a few exceptions, PiCSAR maintains the strongest overall profile. For instance, on MATH500 with $k = 6$, it improves the accuracy of Qwen3-32B from 75.93% (Self-Consistency) to 77.00%.

Our results show that PiCSAR outperforms most existing baselines and datasets, demonstrating consistent improvements across various reasoning tasks. As shown in Appendix C.1, CISC (PiCSAR) consistently outperforms CISC (*p(True)*), indicating its potential for weighting augmentation, but detailed voting strategy analysis remains future work. To verify the statistical significance of our results, we perform the Friedman test (Demšar, 2006), returning a p-value of $\sim 6e^{-17}$, followed by the post-hoc Nemenyi test, which confirms that PiCSAR significantly outperforms all baselines (more in Appendix C.10). *These findings validate our hypothesis that the model's confidence provides more informative clues than frequency-based selection.*

PiCSAR is also sample efficient. PiCSAR with a small sampling budget ($k = 6$) frequently outperforms both Self-Consistency and Self-Certainty at higher sampling budgets ($k = 16, 32$), narrowing the gap to the upper bound by detecting correct reasoning even within a small sample. For instance, Gemma-2-9B Instruct with $k = 6$ (46.53%) outperforms $k = 32$ (43.27%). This indicates that correct reasoning chains are often present in small candidate sets, and that better selection is more important than increased sampling. See Appendix C.7 for details of the upper bound analysis.

Overall, the joint score acts as a paired scoring function: the *reasoning confidence* provides an assessment of plausibility towards its own reasoning, while the *answer confidence*, focused on the final answer, serves as a fine-grained discriminator. This approach yields consistent improvements across evaluated models.

Performance on LRMs. Table 2 reports results from the LRMs. Across 19 out of 21 comparisons, PiCSAR outperforms all baselines. Relative to Self-Consistency, DS-Distill-Llama-3-8B demonstrates substantial improvements on AIME2024 (8.89%) and AIME2025 (8.33%). DS-Distill-Qwen-2.5-7B shows greater improvements compared to Self-Consistency, with an improvement of 12.33% on AIME2024 and of 12.78% on AIME2025. When applied on a relatively more capable model such as Qwen3-8B, PiCSAR increases accuracy by 4.1% and 3.33% on AIME 2024 and AIME 2025, respectively. While improvements on previously evaluated benchmarks (MATH500, SVAMP, GSM8K) yield smaller gains, we observe substantial improvements on GPQA-Diamond: 5.21%, 7.58%, and 5.22% for DS-Distill-Llama-3-8B, DS-Distill-Qwen-2.5-7B, and Qwen3-8B, respectively. These trends mirror those observed with LLMs: gains are most pronounced on challenging datasets where the models' initial baseline accuracies are relatively lower. The Friedman and post-hoc Nemenyi testing additionally confirm that PiCSAR significantly outperforms all baselines (see Appendix C.10).

PiCSAR, validates the information plane principle in §2.3 and provides a scoring method that improves accuracy both for LLMs and LRMs.

Comparison with Trained Reward Models. While our primary baselines consist of training-free BoN methods, a critical question is how PiCSAR compares to explicitly trained verifiers. To establish this, we benchmarked PiCSAR against top-performing reward models on

Rank	Peaks	Sent.	Ratio (%)	Acc. (%)
Llama-3.1-8B				
Highest	1.88	16.4	14.8	53.3
Middle	2.00	22.9	12.8	48.8
Lowest	2.47	64.7	8.6	44.2
Llama-3.1-70B				
Highest	1.80	14.1	15.5	63.7
Middle	1.83	19.9	13.0	60.4
Lowest	3.08	38.4	10.8	59.4
Qwen3-8B				
Highest	1.99	15.8	17.6	73.7
Middle	1.91	17.6	17.0	72.8
Lowest	2.18	26.4	14.2	69.4
Qwen3-32B				
Highest	1.48	11.6	22.4	77.0
Middle	1.57	12.0	19.4	76.8
Lowest	1.76	25.1	16.1	72.6
Gemma-2-9B				
Highest	1.46	8.5	24.5	46.5
Middle	1.38	10.0	19.0	44.0
Lowest	1.20	11.6	14.3	41.6

Table 3: Peak count analysis across different PiCSAR confidence rankings (highest, middle (3rd), and lowest) for different models. **Peaks**: Average number of peaks; **Sent.**: Average sentence count; **Ratio**: Average Peak-to-Sentence ratio; **Acc.**: Model accuracy. Chains in the "Highest" rank consistently show a higher peak density (Ratio) compared to the "Lowest" rank.

the RewardBench (Lambert et al., 2025) leaderboard, specifically *Skywork-Reward-V2-Llama-3.1-8B* and *LMUnit-qwen2.5-72B*. Despite being a completely zero-shot, training-free method, PiCSAR achieves parity with, and in several cases, outperforms these heavily trained reward models across both MATH500 and GSM8K. This confirms that PiCSAR's probabilistic formulation extracts a signal as reliable as explicit preference tuning, but at zero training cost. Detailed empirical results and analysis for this comparison are provided in Appendix E.2.

5 Further Analysis

In our analysis, we study (1) how information density correlates with accuracy; (2) the confidence-accuracy relationship within each model; (3) the robustness of our confidence metric when generation and evaluation are decoupled.

5.1 Sentence-Level Confidence Dynamics as a Proxy for Reasoning Quality

To understand the dynamics of PiCSAR, we analyse the evolution of answer confidence across reasoning chains. For a given reasoning chain r composed of sentences $(r^1, r^2, \ldots, r^m)$ and its corresponding final answer y, we measure how the model's confidence in y changes as it processes more of the reasoning. We compute a sequence of scores, $\log p(y \mid r^{1:j}, x)$, for each partial reasoning prefix $r^{1:j}$, where j ranges from 1 to m. To capture the characteristics of these confidence sequences, we rank the responses by PiCSAR scoring function into three groups (highest, middle, lowest), and analyse the "peakiness" of the confidence trajectory within each group. We define a *peak* as a sentence where the confidence $\log p(y \mid r^{1:j}, x)$ exceeds the 95th percentile of all sentence-level scores observed across reasoning chains with the correct answer for that particular problem. The *peak-to-sentence ratio* is the peak count divided by the total sentences. We term this *information density*: the proportion of reasoning sentences contributing meaningfully to answer confidence.

Table 3 shows: (1) Higher peak-to-sentence ratio aligns with higher accuracy across different models, showing that *reasoning chains that lead to the correct answer tend to have higher information density*. For instance, Llama-3.1-8B achieves 53.33% accuracy with a 14.75% ratio in the highest-scoring group, compared to 44.20% with only 8.58% in the lowest; (2) *Longer reasoning chains do not necessarily improve accuracy*. The lowest-ranked responses are substantially longer yet less accurate. For example, Llama-3.1-8B averages 64.72 sentences with 44.20% accuracy in the lowest group, versus 16.43 sentences with 53.33% accuracy in the highest group. This observation aligns with recent findings of inverse scaling in test-time compute (Chen et al., 2025; Wu et al., 2025; Hassid et al., 2025; Ghosal et al., 2025; Gema et al., 2025a), showing that solely extended reasoning length does not guarantee improved performance.

As unnormalised PiCSAR naturally rewards these high-density, convergent trajectories, it serves as our recommended default. Length normalisation (PiCSAR-N) is typically only necessary when evaluating weaker models that are highly prone to "verbose hallucinations", where the model accumulates massive negative log-probabilities through unproductive, circular generation rather than meaningful reasoning. We provide a comprehensive sentence-level trajectory analysis detailing the exact criteria and decision boundary for enabling length normalisation in Appendix C.6 and Appendix D.

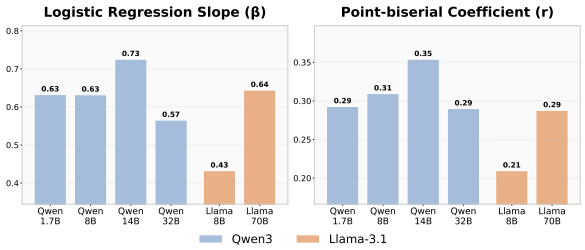


Figure 4: Calibration summary for Qwen3 and Llama-3.1-8B models. We show that the β and r coefficients are consistently positive across all models.

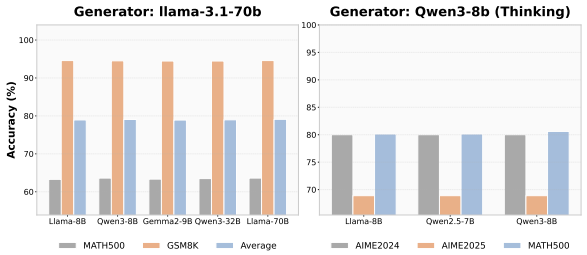


Figure 5: Decoupling analysis for Llama-3.1-70B and Qwen3-8B (Thinking Enabled) as M_{gen}, with various M_{eval}, showing performance remains similar when different models are used to estimate $\log p(y \mid r, x)$.

5.2 Intra-model Confidence Duality

In this section, we investigate the reliability of PiCSAR for predicting correctness within individual models (*intra-model reliability analysis*). We further examine whether these confidence scores remain comparable across different models (*inter-model variance analysis*) in Appendix J. We fit regressions for the Qwen and Llama families (Figure 4), with correctness (correct/incorrect) as the dependent variable and the answer confidence score as the independent variable. This approach allows us to interpret the regression slope (β), which represents the incremental change in log-odds of correctness per unit increase in confidence score.

We find that the β is consistently positive across all model sizes, consistent with prior findings Huh et al. (2024); Goel et al. (2025) of a strong positive relationship between confidence scores and their likelihood of being correct. For example, Qwen3-14B shows a β of 0.7255, implying that each unit increase in log-probability more than doubles the odds of correctness ($e^{0.7255} \approx 2.07$). The Point-Biserial Correlation Coefficient further confirms the positive relationship by measuring the linear association between binary correctness and continuous confidence. *These findings show that PiCSAR serves as a reliable predictor of correctness within each model.* See Appendix I for more details.

5.3 Confidence Portability: Decoupling Generation from Evaluation

Having established the properties of the confidence signal within a single model, we extend our analysis to multi-model scenarios, evaluating confidence signal robustness when generation and evaluation are decoupled. This decoupling is motivated by practical system design, where one might use a costly API model for reasoning confidence, while relying on a smaller local model for answer confidence estimation. In this *decoupled* setting, the model that generates the reasoning chain (M_{gen}) differs from the model that evaluates the answer confidence (M_{eval}). The scoring function for a chain r_i generated by M_{gen} becomes:

$$\text{Score}(r_i, y_i) = \underbrace{\log p(r_i \mid x; M_{\text{gen}})}_{\text{Generated by } M_{\text{gen}}} + \underbrace{\log p(y_i \mid \langle a \rangle, r_i, x; M_{\text{eval}})}_{\text{Evaluated by } M_{\text{eval}}}. \quad (4)$$

We test this by having M_{gen} generate reasoning chains, and various models acting as M_{eval}. For LRMs, the base instruction tuned model is used as M_{eval}. Results in Figure 5 and Appendix A show that overall accuracy remains largely unaffected under this decoupling, with only minor degradation even when M_{eval} is a significantly smaller model than M_{gen}. For instance, accuracy remains similar when M_{gen} is generated by Llama-3.1-70B, while M_{eval} is estimated with other smaller models. This suggests that the answer confidence term, $\log p(y \mid r, x)$, is not merely a model-specific artefact but functions as a more portable measure of the logical entailment between a given reasoning chain and its conclusion, enabling flexible and computationally efficient answer confidence prediction.

6 Related Work

LLM Reasoning LLM reasoning abilities has gained significantly on complex tasks (Li et al., 2025; Muennighoff et al., 2025). While CoT reasoning improves performance (Wei et al., 2022; Leang et al., 2025a), subsequent work introduced hierarchical reasoning phases: multi-path exploration (Yao et al., 2023; Guan et al., 2025), step verification (Lightman et al., 2024; Leang et al., 2025b), iterative refinement (Madaan et al., 2023), and analysis and implementation of repeated inference (Levi, 2025). These techniques are computationally prohibitive for LRMs (Team et al., 2025; Yang et al., 2025a), which produce long, unstructured outputs.

BoN. BoN is an alignment-via-inference method that optimises outputs with a scoring function (Charniak and Johnson, 2005; Stiennon et al., 2020; Amini et al., 2024). With scale-time inference, LLMs benefit from generating multiple samples and selecting the best via reward models (Snell et al., 2024). Due to their training cost, reward models are often replaced by training-free methods such as Self-Consistency and its variants (Wan et al., 2024; Lyu et al., 2025).

Sampling and Reranking. Reranking improves generation quality (Adiwardana et al., 2020; Shen et al., 2021), often via trained verifiers to re-rank candidates, outperforming fine-tuning (Cobbe et al., 2021; Guan et al., 2025). Confidence estimation for re-ranking has been explored via sample agreement (Kuhn et al., 2023; Manakul et al., 2023; Tian et al., 2024; Simhi et al., 2025), or prompting models to verbalise confidence (Tian et al., 2023; Kadavath et al., 2022).

7 Conclusion

We introduced PiCSAR, a sample-efficient, training-free scoring function for BoN sampling that selects a reasoning chain by maximising a score decomposed into reasoning and answer confidence. PiCSAR yields consistent gains across models and datasets, narrowing the gap to oracle performance while requiring only $k = 6$ samples to outperform baselines using $k = 32$. The answer confidence component can be estimated by different models than the one used for generation, enabling flexible and efficient deployment. At the trajectory level, peak-count-to-sentence ratios correlate with accuracy, showing that reasoning chains leading to correct answers are more information-dense. Overall, PiCSAR offers a promising probabilistic confidence route to reasoning selection.

Limitations

PiCSAR targets domains with well-defined reasoning structures and definitive answers, such as *mathematical and scientific problem-solving*. We view this scope as both deliberate and essential: these domains represent a substantial class of high-value reasoning tasks where precision is important. Furthermore, restricting our analysis to these settings enables a rigorous evaluation of confidence calibration, a task that remains difficult in open-ended domains – could be characterised by ambiguity and multiple valid solutions. This controlled environment allows us to validate the efficacy of model confidence as a selection metric without the confounding factors of subjective evaluation.

Extending PiCSAR to open-ended generation remains an important avenue for future research. To address the lack of definitive answer boundaries in such tasks, a promising direction is to augment the probabilistic framework with learned reward models for answer evaluation. We believe this adaptation could extend the reliability benefits of PiCSAR beyond fixed-format problems, offering a pathway toward robust reasoning in broader, general-purpose applications.

Acknowledgements

We thank the anonymous reviewers and area chairs for their helpful comments and feedback. We also thank Waylon Li and Adi Simhi for their valuable feedback. Lastly, we are grateful for the compute resources provided to us by the University of Edinburgh (Edinburgh International Data Facility), and UKRI (Isambard AI service, University of Bristol).

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and 1 others. 2020. Towards a human-like open-domain chatbot. *ArXiv preprint*, abs/2001.09977.

Afra Amini, Tim Vieira, Elliott Ash, and Ryan Cotterell. 2024. Variational best-of-n alignment. *ArXiv preprint*, abs/2407.06057.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. 2025. Matharena: Evaluating LLMs on uncontaminated math competitions. *ArXiv preprint*, abs/2505.23281.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, Michigan. Association for Computational Linguistics.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. 2023a. TheoremQA: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7889–7901, Singapore. Association for Computational Linguistics.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi

Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Do NOT think that much for 2+3=? on the overthinking of long reasoning models. In *Forty-second International Conference on Machine Learning*.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023b. Universal self-consistency for large language model generation. *ArXiv preprint*, abs/2311.17311.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *ArXiv preprint*, abs/2110.14168.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The Llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, and 1 others. 2023. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *ArXiv preprint*, abs/2312.09244.

Aryo Pradipta Gema, Alexander Hägele, Runjin Chen, Andy Arditi, Jacob Goldman-Wetzler, Kit Fraser-Taliente, Henry Sleight, Linda Petrini, Julian Michael, Beatrice Alex, and 1 others. 2025a. Inverse scaling in test-time compute. *ArXiv preprint*, abs/2507.14417.

Aryo Pradipta Gema, Chen Jin, Ahmed Abdulaal, Tom Diethe, Philip Alexander Teare, Beatrice Alex, Pasquale Minervini, and Amrutha Saseendran. 2025b. DeCoRe: Decoding by contrasting retrieval heads to mitigate hallucinations. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10003–10039, Suzhou, China. Association for Computational Linguistics.

Soumya Suvra Ghosal, Souradip Chakraborty, Avinash Reddy, Yifu Lu, Mengdi Wang, Dinesh Manocha, Furong Huang, Mohammad Ghavamzadeh, and Amrit Singh Bedi. 2025. Does thinking more always help? understanding test-time scaling in reasoning models. *ArXiv preprint*, abs/2506.04210.

Shashwat Goel, Joschka Struber, Ilze Amanda Auzina, Karuna K Chandra, Ponnurangam Kumaraguru, Douwe Kiela, Ameya Prabhu, Matthias Bethge, and Jonas Geiping. 2025. Great models think alike and this undermines AI oversight. *ArXiv preprint*, abs/2502.04313.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *ArXiv preprint*, abs/2407.21783.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rStar-Math: Small LLMs can master math reasoning with self-evolved deep thinking. *ArXiv preprint*, abs/2501.04519.

Daya Guo, Dejian Yang, Haowei Zhang, and 1 others. 2025. Deepseek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Nature*, 645:633–638.

Michael Hassid, Gabriel Synnaeve, Yossi Adi, and Roy Schwartz. 2025. Don't overthink it. preferring shorter thinking chains for improved llm reasoning. *ArXiv preprint*, abs/2505.17813.

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

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J Foster. 2025. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment. *ArXiv preprint*, abs/2503.21878.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. The platonic representation hypothesis. *ArXiv preprint*, abs/2405.07987.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. GPT-4o system card. *ArXiv preprint*, abs/2410.21276.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *ArXiv preprint*, abs/2207.05221.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. RewardBench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1755–1797, Albuquerque, New Mexico. Association for Computational Linguistics.

Joshua Ong Jun Leang, Aryo Pradipta Gema, and Shay B Cohen. 2025a. CoMAT: Chain of mathematically annotated thought improves mathematical reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20245–20274, Suzhou, China. Association for Computational Linguistics.

Joshua Ong Jun Leang, Giwon Hong, Wenda Li, and Shay B Cohen. 2025b. Theorem Prover as a Judge for Synthetic Data Generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29941–29977, Vienna, Austria. Association for Computational Linguistics.

Noam Itzhak Levi. 2025. A simple model of inference scaling laws. In *Forty-second International Conference on Machine Learning*.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, and 1 others. 2025. From system 1 to system 2: A survey of reasoning large language models. *ArXiv preprint*, abs/2502.17419.

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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Qing Lyu, Kumar Shridhar, Chaitanya Malaviya, Li Zhang, Yanai Elazar, Niket Tandon, Marianna Apidianaki, Mrinmaya Sachan, and Chris Callison-Burch. 2025. Calibrating large language models with sample consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19260–19268.

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. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Ning Miao, Yee Whye Teh, and Tom Rainforth. 2024. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. 2024. Controlled decoding from language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 36486–36503. PMLR / OpenReview.net.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *ArXiv preprint*, abs/2501.19393.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

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

Jianhao Shen, Yichun Yin, Lin Li, Lifeng Shang, Xin Jiang, Ming Zhang, and Qun Liu. 2021. Generate & rank: A multi-task framework for math word problems. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2269–2279,

Punta Cana, Dominican Republic. Association for Computational Linguistics.

Adi Simhi, Jonathan Herzig, Itay Itzhak, Dana Arad, Zorik Gekhman, Roi Reichart, Fazl Barez, Gabriel Stanovsky, Idan Szpektor, and Yonatan Belinkov. 2025. Hack: Hallucinations along certainty and knowledge axes. *ArXiv preprint*, abs/2510.24222.

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

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. Confidence improves self-consistency in LLMs. *ArXiv preprint*, abs/2502.06233.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv preprint*, abs/2408.00118.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi K1.5: Scaling reinforcement learning with LLMs. *ArXiv preprint*, abs/2501.12599.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D. Manning, and Chelsea Finn. 2024. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Jean-Francois Ton, Muhammad Faaiz Taufiq, and Yang Liu. 2024. Understanding chain-of-thought in LLMs through information theory. *ArXiv preprint*, abs/2411.11984.

Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. 2024. Reasoning aware self-consistency: Leveraging reasoning paths for efficient LLM sampling. *ArXiv preprint*, abs/2408.17017.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2023a. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. *ArXiv preprint*, abs/2312.08935.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yuyang Wu, Yifei Wang, Tianqi Du, Stefanie Jegelka, and Yisen Wang. 2025. When more is less: Understanding chain-of-thought length in LLMs. *ArXiv preprint*, abs/2502.07266.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *ArXiv preprint*, abs/2505.09388.

Sohee Yang, Sang-Woo Lee, Nora Kassner, Daniela Gottesman, Sebastian Riedel, and Mor Geva. 2025b. How well can reasoning models identify and recover from unhelpful thoughts? *ArXiv preprint*, abs/2506.10979.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

A Additional Results for Decoupled Confidence Estimation

In this section, we provide supplementary evidence that the decoupled confidence estimation experiments introduced in §5.3 are portable across distinct evaluator models. This analysis aims to strengthen the claim that the answer-confidence term, $\log p(y \mid r, x)$, does not depend on the specific evaluator used.

Based on Figure 6a, switching the evaluator model, M_{eval} while holding the reasoning distribution fixed yields a similar accuracy across datasets. This observation shows that the answer-confidence term, $\log p(y \mid r, x)$, is highly portable, allowing small-scale LLMs to reliably evaluate the reasoning chains of larger models.

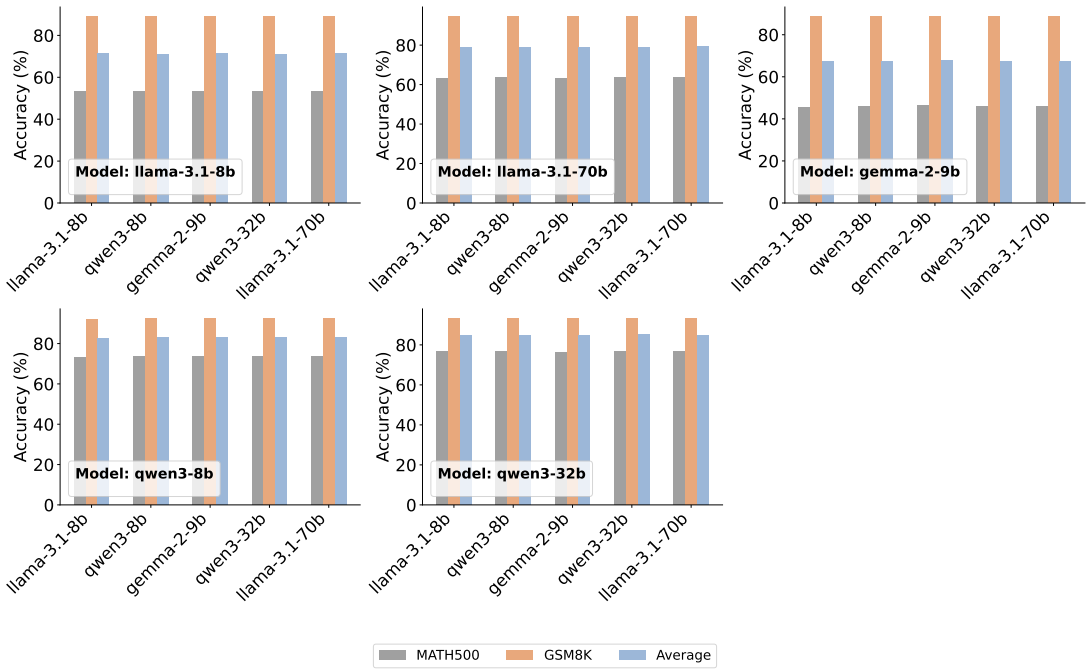
When examining LRMs, we observe the same qualitative pattern (shown in Figure 6b), indicating that the phenomenon generalises across models. This reinforces the hypothesis that decoupled confidence estimation captures a stable property of the reasoning process itself, rather than an artefact of the evaluator model.

B Additional Implementation Details

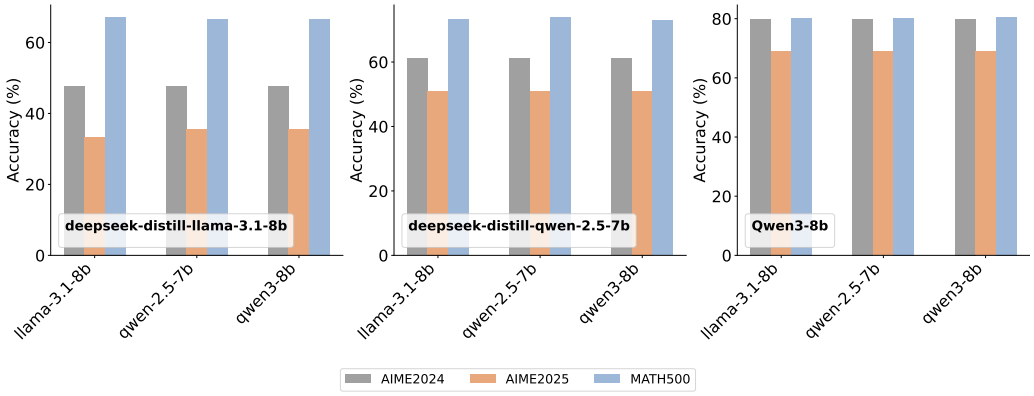
Sampling and Decoding. For sampling-based methods, we use $k \in \{6, 32\}$ reasoning traces for smaller models and $k \in \{6, 16\}$ for the larger Llama-3.1-70B and Qwen3-32B models, due to computational constraints. For all the models, we apply a hyperparameter of temperature $= 0.7$ and top-p $= 0.6$. The greedy decoding (temperature $= 0$, top-p $= 1.0$) baseline corresponds to $k = 1$, for which we report Pass@1 accuracy. For specialised LRMs, we use $k = 6$ uniformly across all methods due to computational constraints. Since LRMs are not typically evaluated using greedy decoding, we follow the approach of Yang et al. (2025a), which is a temperature of 0.6, top-k of 20 and top-p $= 0.95$, reporting the average accuracy across k samples. For all our baselines except greedy decoding, we evaluate three times with the standard error reported. For LLMs, we cap the maximum token budget at 8,096 tokens. For LRMs, we follow the configuration of Yang et al. (2025a), using a maximum output length of 32,768 tokens, except for AIME'24 and AIME'25, where we extend the budget to 38,912 tokens to ensure sufficient reasoning space.

Baselines and Hyperparameters. We compare PiCSAR against a range of decoding, confidence and re-ranking baselines.

- **Greedy Decoding** As a deterministic decoding strategy, greedy decoding selects at each step the token with the highest conditional probability. Unlike greedy decoding, which selects a single high-probability continuation, PiCSAR evaluates multiple full reasoning trajectories and ranks them using joint reasoning-and-answer log-likelihood, enabling selection of the most globally probable chain.
- **Self-Consistency** (SC; Wang et al. 2023b). This method samples k reasoning chains and aggregates predictions via majority voting on the final answer. In cases where multiple answers receive equal support, we break ties by selecting one at random. While SC relies purely on majority voting over final answers, PiCSAR incorporates the full reasoning chain's token-level likelihood along with answer confidence, allowing it to prefer coherent but minority reasoning paths that SC would discard.
- **Universal Self-Consistency** (USC; Chen et al. 2023b). We include USC only for LLMs under k=6 sampling, as prompt and context length restrictions prevent its application in the LRM setting. We use the prompting strategy proposed in Chen et al. (2023b). Unlike USC, which asks the model to internally judge "consistency" among samples, PiCSAR uses a probabilistic, model-agnostic scoring function based directly on log-likelihoods of reasoning and answers, avoiding USC's reliance on model self-evaluation and context-window limits.
- **Self-Certainty** (Kang et al., 2025). This method applies KL-divergence-based confidence scores, aggregated via Borda voting with parameter p=0.5. It provides a probabilistic variant of self-consistency, where each candidate's confidence distribution informs the re-ranking process. Instead of re-ranking chains with KL-based self-estimated correctness like Self-Certainty, PiCSAR scores each candidate through the true generative probabilities of its entire reasoning path and answer
- ***P(True)*** (Kadavath et al., 2022). This method prompts the model to evaluate whether the answer or reasoning is *True* or *False*, then parses the probability of the response. While P(True) extracts a scalar correctness probability from a



(a) Decoupling plot by using various LLMs to evaluate $p(y \mid r, x)$ across a particular model reasoning chain, $p(r \mid x)$. Each subplot represents a M_{gen}, and the *x-axis* represents various M_{eval}. The results remain similar when M_{eval} varies, even with smaller models predicting larger M_{gen}.



(b) Decoupling plot by using various LRMs to evaluate $p(y \mid r, x)$ across a particular model reasoning chain, $p(r \mid x)$. Each subplot represents a M_{gen}, and the *x-axis* represents various M_{eval}. The results remain similar when M_{eval} varies, even with smaller models predicting larger M_{gen}.

Figure 6: Decoupling analysis for LLMs and LRMs (Thinking Enabled) as M_{gen}, with various M_{eval}, showing performance remains similar when different models are used to estimate $\log p(y \mid r, x)$.

meta-prompt, PiCSAR leverages the actual likelihood structure of the model's forward pass, combining reasoning and answer probabilities without relying on verbalized or poorly calibrated self-judgments.

- **CISC** (Taubenfeld et al., 2025). This method aggregates multiple sampled reasoning paths by weighting each path's vote with the model's own estimated correctness. For a fair comparison, we compare CISC with PiCSAR as estimated correctness, termed CISC (PiCSAR), with CISC (*P(True)*, which originally proposed, in Appendix C.1.

We have summarised the novelty of PiCSAR against other baselines in Table 4.

Baseline Restrictions. Due to context length constraints, USC can only handle a limited number of samples and is therefore evaluated exclusively in the LLM setting with k=6, and excluded from all LRM experiments.

Ablations. To disentangle the contributions of the two terms in our joint objective, we introduce

Method	SC	USC	Self-Cert.	PiCSAR
Full Reasoning Chain	✓	✓		✓
Model Confidence		✓	✓	✓
Computationally Efficient	✓	×*	✓	✓
Smaller Model Capable	✓		✓	✓

*Due to context length

Table 4: Comparison of different baselines with our proposed method across several dimensions.

single-term ablations. *Reasoning Confidence* ranks candidates solely by $\log p(r \mid x)$, favouring plausible reasoning traces. *Answer Confidence* instead ranks by $\log p(y \mid r, x)$, prioritising certainty in the final answer given the reasoning path.

Framework and Hardware. All experiments are conducted using the vLLM framework (Kwon et al., 2023). All experiments are conducted on 2–4 NVIDIA H100 GPUs (80GB). Results are reported as averages over independent evaluation runs to ensure robustness.

Prompt. For the reasoning confidence $\log p(r \mid x)$ generation, we utilise the following prompt:

```
You are a helpful AI Assistant that provides well-reasoned and detailed responses. Think step by step and provide the final answer in the form of 'The final answer is: [answer]'. Decompose and break down your reasoning into smallest possible steps (Do not combine multiple inferences in one step), and do label your steps very clearly with 'Step 1... \n\n Step 2... \n\n Step 3.... \n\n..... \n\n Step N-1..... \n\n Step N \n\n The final answer is: [answer]'.
```

For predicting answer confidence $\log p(y \mid r, x)$, we follow a similar method to (Ton et al., 2024) but without training. Specifically, we use the prompt template $\langle a \rangle$ with 5-shot learning:

```
You are a helpful assistant. When you see a potential partial reasoning followed by '<sep>', output the final answer.
```

B.1 Analyis of Prompts

To verify that the observed improvements are not attributable to the explicit instruction prompt (see (3)), we evaluated several alternative prompt formulations on the Llama-3.1-8B model. Using the MATH500 benchmark, we compared the resulting answer-confidence estimates across prompts.

```
Prompt 1: "You are a helpful assistant. When you see a potential partial reasoning followed by '<sep>', output the final answer. Here are some examples" + system_contents + "You are not allowed to provide any redundant symbols at for the final answer, including '#', '/', '$', '**' or others. Please only provide numbers as the final answer."
```

```
Prompt 2 (original prompt): "You are a helpful assistant. When you see a potential partial reasoning followed by '<sep>', output the final answer. Here are some examples"
```

```
Prompt 3: "You are a helpful assistant. By providing the partial reasoning, output the final answer directly without any additional texts."
```

```
Prompt 4: "You are a helpful assistant. Based on the reasoning provided, output the final answer directly without any additional texts. Only Provide the final answer."
```

```
Prompt 5: "You are a helpful assistant. Provide the final answer directly without any additional texts (only the final answer) based on the partial reasoning."
```

Prompt	Accuracy
Prompt 1	54.60%
Prompt 2	54.00%
Prompt 3	54.20%
Prompt 4	54.40%
Prompt 5	54.40%

Table 5: Performance of PiCSAR on Llama-3.1-8B on MATH500 with different prompts for answer-confidence extraction.

Our results in Table 5 show that changes in prompt phrasing have minimal influence on model performance. This suggests that, although the instructional content of a prompt remains essential for eliciting the final answer, the precise wording

Method	SVAMP		GSM8K		MATH500		TheoremQA	
	$k=6$	$k=16/32$	$k=6$	$k=16/32$	$k=6$	$k=16/32$	$k=6$	$k=16/32$
Gemma-2-9B-Instruct								
CISC (*p(True)*	89.22±0.22	88.67±0.38	88.89±0.26	89.14±0.15	46.87±0.33	47.67±0.07	17.09±0.43	17.45±0.12
PiCSAR	89.00±0.38	91.02±0.59	88.66±0.11	88.99±0.20	46.53±0.29	47.13±0.13	18.62±0.39	18.88±0.54
CISC (PiCSAR)	**91.89±0.22**	**92.33±0.19**	**91.85±0.20**	**92.43±0.22**	**51.33±0.07**	**52.13±0.29**	**21.02±0.58**	**23.16±0.39**
Upper Bound	24.32±0.49	32.40±0.20	93.44±0.09	95.60±0.04	58.47±0.27	66.67±0.47	55.22±1.10	82.49±1.02
Llama-3.1-8B-Instruct								
CISC (*p(True)*)	91.44±0.48	92.78±0.29	91.17±0.18	91.91±0.49	54.93±0.41	58.20±0.42	18.03±0.73	39.38±18.91
PiCSAR	91.78±0.11	93.44±0.89	89.09±0.13	89.98±0.23	53.33±0.73	53.87±0.70	20.08±0.43	19.72±0.39
CISC (PiCSAR)	**94.33**±0.33	**96.22**±0.11	**93.98**±0.14	**94.23**±0.08	**62.47**±0.07	**62.40**±0.50	**22.71**±0.25	**41.50**±17.34
Upper Bound	96.78±0.11	99.11±0.11	96.15±0.07	98.18±0.04	72.80±0.23	82.20±0.60	28.20±0.32	37.846±1.13
Qwen3-8B (Non-thinking)								
CICS (*p(True)*)	94.33±0.00	94.56±0.11	93.80±0.13	94.05±0.14	77.20±0.20	77.93±0.24	31.24±0.04	32.75±0.45
PiCSAR	93.56±0.22	95.13±0.22	92.33±0.13	93.22±0.08	73.67±0.24	73.40±0.13	29.76±0.57	29.17±0.64
CICS (PiCSAR)	**95.11±0.11**	**95.67±0.19**	**94.89±0.14**	**95.22±0.12**	**79.80±0.40**	**79.60±0.42**	**36.46±0.04**	**36.32±0.04**
Upper Bound	96.33±0.67	97.89±0.11	95.52±0.00	96.84±0.03	81.13±0.44	83.53±0.24	34.94±0.00	40.03±0.35
Llama-3.1-70B-Instruct								
CISC (*p(True)*)	94.22±0.22	94.11±0.11	94.68±0.00	95.09±0.09	65.07±1.05	66.27±0.29	28.07±0.68	29.41±0.12
PiCSAR	94.10±0.11	95.58±0.22	94.58±0.03	94.81±0.13	63.67±1.51	64.07±0.87	27.84±0.19	26.73±0.27
CISC (PiCSAR)	**96.78**±0.11	**96.44**±0.11	**95.90**±0.08	**96.03**±0.11	**69.60**±0.31	**70.80**±0.76	**31.91**±0.31	**31.59**±0.27
Upper Bound	97.22±0.22	97.78±0.22	96.91±0.03	97.44±0.03	77.07±0.47	81.67±0.18	40.70±0.20	43.47±0.18
Qwen3-32B (Non-thinking)								
CICS (P-True)	94.33±0.00	94.56±0.11	93.80±0.13	94.05±0.14	77.20±0.20	77.93±0.24	31.24±0.04	32.75±0.45
PiCSAR	93.22±0.22	93.55±0.33	93.90±0.28	93.88±0.22	77.00±0.18	75.93±0.13	31.46±0.04	31.42±0.27
CICS (PiCSAR)	**95.11**±0.11	**95.67**±0.19	**94.89**±0.14	**95.22**±0.12	**79.80**±0.40	**79.60**±0.42	**36.46**±0.04	**36.32**±0.04
Upper Bound	96.78±0.11	98.00±0.00	96.28±0.13	96.99±0.07	82.27±0.13	83.73±0.07	39.76±0.00	42.93±0.12

Table 6: **Performance comparison on benchmarks across CISC (*p(True)*) and CISC (PiCSAR) on LLMs.** Values represent mean accuracy ± standard error over three independent evaluation runs. **Bold** indicates the best-performing method per column based on the mean accuracy. Sampling parameters: $k = \{6, 32\}$ for Gemma-2-9B, Llama-3.1-8B, and Qwen3-8B; $k = \{6, 16\}$ for Llama-3.1-70B and Qwen3-32B.

plays only a limited role in shaping the model's behaviour.

C Further Experimental Results and Ablation Studies

C.1 Comparison between CISC (p(True)) and CISC (PiCSAR)

Based on Table 6, PiCSAR shows a great performance when integrated with weightage voting on CISC (Taubenfeld et al., 2025), consistently improving baseline CICS (*p(True)*) metrics across all evaluated methods. This indicates that PiCSAR functions effectively both as a standalone selection mechanism and as an augmentation to existing weighting schemes. While these findings suggest promising direction for performance optimisation, this lies beyond the current research scope.

C.2 Component Analysis and Main Results Breakdown

In this section, we first provide a detailed breakdown of the experimental results for all methods, as summarised in Table 7, where we also show the performance of PiCSAR-N, a length-normalised variant of our primary method. Finally, we present ablation studies on LRMs in Table 8. We compare three primary approaches: *Reasoning Confidence* ($\log p(r \mid x)$), *Answer Confidence* ($\log p(y \mid r, x)$), and our main method, *PiCSAR* (the joint probability).

Across the majority of benchmarks and model families presented in Table 7, we generally observe that PiCSAR outperforms its individual components. This pattern underscores the benefit of jointly considering the likelihood of both the reasoning process and the final answer. However, there are specific instances where relying solely on answer confidence, $\log p(y \mid r, x)$, achieves comparable or slightly better results (e.g., Gemma-2-9B and Qwen3-32B on GPQA-Diamond for $k = 32$), highlighting that answer confidence remains a strong and competitive signal on its own.

C.3 Length-Normalised Variant: PiCSAR-N

As introduced in the main paper, we proposed a variant of our method, PiCSAR-N, which applies length normalisation to the reasoning confidence

term. The scoring function for PiCSAR-N is defined as:

$$\text{Score}(r, y) = \left[\frac{1}{N} \log p(r \mid x)\right] + \log p(y \mid \langle a \rangle, r, x), \tag{5}$$

where N is the number of tokens in the reasoning chain r. This normalisation is intended to mitigate any potential length bias, which might unfairly penalise longer reasoning paths.

C.4 Analysis between Token Length, PiCSAR score, and Model Performance

Figure 7a shows that correct instances predominantly cluster in regions of high probability and short sequence length, indicating that concise reasoning is strongly associated with higher quality. This pattern is reinforced by Figure 7b, which demonstrates a consistent decline in accuracy as sequence length grows. Together, the two figures highlight that shorter, more confident reasoning trajectories tend to yield more accurate performance.

C.5 Ablation Studies on LLMs and LRMs

The results for PiCSAR-N are included in Table 7 and Table 8. As shown, both PiCSAR and PiCSAR-N consistently surpass the other baselines, including their corresponding reasoning confidence metrics (with and without normalisation). The performance difference between PiCSAR and PiCSAR-N is not consistently in one direction; each variant excels on different model-dataset combinations. For instance, PiCSAR-N shows stronger performance with Gemma-2-9B on MATH500 ($k = 6$) and GPQA-Diamond, whereas the non-normalised PiCSAR is clearly superior for Llama-3.1-8B across most settings. This suggests that the utility of length normalisation may depend on model-specific characteristics, such as tendencies towards verbosity.

Based on Table 7, we also observe that 20/40 results of the length-normalised (PiCSAR-N) versions outperform the non-length normalised versions (PiCSAR), demonstrating that length-normalisation does not perform worse than the non-length normalised version. This suggests that length normalisation is not detrimental and does not consistently weaken PiCSAR.

We further conducted ablation studies on LRMs, with results reported in Table 8. Here, we compare PiCSAR and PiCSAR-N against both standard and normalised reasoning confidence, as well as answer confidence. The results confirm that our joint probability methods, PiCSAR and PiCSAR-N, consistently achieve top performance, similar to the findings with LLMs. Interestingly, we observe that maximising answer confidence alone yields strong results, sometimes comparable to PiCSAR, particularly on the DS-Distill-llama-3-8B model. This reinforces the value of the answer confidence signal while highlighting the general effectiveness of PiCSAR's approach in combining both reasoning and answer confidence.

C.6 Further Analysis on Length-Normalised Variant: PiCSAR-N

In this section, we clarify the distinctions between PiCSAR and its length-normalised counterpart, PiCSAR-N, establishing empirical evidence for when length normalisation should be applied.

As shown in Table 7, the performance gap between PiCSAR and PiCSAR-N is generally marginal, with neither variant strictly dominating across all model-task configurations. We introduce PiCSAR-N primarily as an ablation to confirm that PiCSAR's strong performance is not merely an artifact of systematically penalising longer generations. Therefore, we recommend the unnormalised PiCSAR as the default selection strategy. Our empirical analysis (see§ 5.1) suggests that correct reasoning chains exhibit high "information density". They accumulate log-probability mass efficiently as they converge toward the final answer. The unnormalised joint log-likelihood naturally favours reasoning paths that are both highly probable and structurally concise, effectively penalising indirect reasoning paths.

Conversely, PiCSAR-N proves beneficial primarily for weaker models prone to "verbose hallucinations", instances where a model generates locally plausible but logically stagnant text that accumulates massive negative log-probabilities strictly due to sequence length. For highly capable reasoners (*e.g.,* Llama-3.1, Qwen3), the unnormalised score remains highly robust, as these models' sequence-level log-probabilities serve as well-calibrated proxies for both logical coherence and problem-solving efficiency.

Method	SVAMP		GSM8K		MATH500		GPQA-Diamond	
	$k=6$	$k=16/32$	$k=6$	$k=16/32$	$k=6$	$k=16/32$	$k=6$	$k=16/32$
Gemma-2-9B-Instruct								
Reasoning Confidence	88.66±0.33	89.67±0.49	88.51±0.05	88.46±0.25	45.87±0.47	45.87±0.68	30.64±0.45	32.32±1.52
Answer Confidence	89.66±0.33	89.02±0.59	88.05±0.17	87.04±0.05	46.47±0.66	46.33±0.18	34.01±2.65	**38.22**±1.76
Reasoning confidence (normalised)	89.56±0.44	90.22±0.29	88.76±0.26	**89.45**±0.20	46.33±0.67	46.47±0.18	29.80±1.91	27.95±2.15
PiCSAR	89.00±0.38	**91.02**±0.59	88.66±0.11	88.99±0.20	46.53±0.29	**47.13**±0.13	32.32±0.51	34.01±1.94
PiCSAR-N	**89.67**±0.19	89.22±0.29	**88.91**±0.12	89.27±0.11	**46.60**±0.92	46.93±0.18	**35.35**±1.62	38.05±1.90
Upper Bound	93.44±0.22	95.67±0.38	93.44±0.09	95.60±0.04	58.47±0.27	66.67±0.47	55.22±1.10	82.49±1.02
Llama-3.1-8B-Instruct								
Reasoning Confidence	91.56±0.11	92.10±0.84	88.89±0.09	89.67±0.27	53.07±0.37	51.53±0.35	29.12±1.02	32.49±2.92
Answer Confidence	89.11±0.29	90.44±0.95	86.84±0.20	86.69±0.04	49.27±0.64	50.20±0.35	28.62±0.73	29.46±2.63
Reasoning confidence (normalised)	90.22±0.11	90.67±0.69	88.38±0.23	86.10±0.08	50.67±0.47	47.13±1.39	22.05±0.89	18.35±0.84
PiCSAR	**91.78**±0.11	**93.44**±0.89	**89.09**±0.13	**89.98**±0.23	**53.33**±0.73	**53.87**±0.70	29.80±1.34	**33.67**±3.06
PiCSAR-N	90.22±0.48	92.22±0.29	88.59±0.18	89.33±0.42	51.53±0.48	51.60±0.42	**30.81±0.87**	30.64±1.61
Upper Bound	96.78±0.11	99.11±0.11	96.15±0.07	98.18±0.04	72.80±0.23	82.20±0.60	65.82±1.50	92.76±0.73
Qwen3-8B (Non-thinking)								
Reasoning Confidence	92.78±0.11	94.34±0.33	92.26±0.13	92.31±0.03	73.53±0.24	72.53±0.48	45.96±1.01	43.77±1.21
Answer Confidence	93.45±0.19	94.02±0.40	93.22±0.03	92.94±0.17	71.07±0.41	71.20±0.76	**51.01**±1.52	43.43±2.53
Reasoning Confidence (normalised)	93.33±0.00	93.67±0.69	92.79±0.00	92.61±0.20	71.93±0.71	69.27±0.44	43.43±0.51	38.05±1.78
PiCSAR	93.56±0.22	**95.13**±0.22	92.33±0.13	93.22±0.08	73.67±0.24	**73.40**±0.13	46.98±1.01	43.69±1.26
PiCSAR-N	**94.44±0.11**	94.56±0.59	**93.69**±0.00	**93.77±0.13**	**73.80±0.20**	72.13±0.98	47.98±1.01	**44.95±0.58**
Upper Bound	96.33±0.67	97.89±0.11	95.52±0.00	96.84±0.03	81.13±0.44	83.53±0.24	76.26±1.62	86.36±0.29
Llama-3.1-70B-Instruct								
Reasoning Confidence	**94.44**±0.11	94.80±0.19	94.46±0.08	93.62±0.18	63.47±1.35	63.00±0.10	43.94±2.62	45.96±2.54
Answer Confidence	93.89±0.22	94.67±0.38	94.10±0.25	94.68±0.23	59.40±1.30	60.07±1.09	45.12±0.45	42.26±1.78
Reasoning Confidence (normalised)	93.33±0.38	93.89±0.22	93.37±0.03	93.34±0.26	65.60±0.60	65.13±0.13	40.07±1.87	37.04±0.89
PiCSAR	94.10±0.11	**95.58**±0.22	**94.58**±0.03	**94.81**±0.13	63.67±1.51	64.07±0.87	46.91±2.65	**46.46**±2.59
PiCSAR-N	**94.44±0.11**	94.56±0.59	94.07±0.00	94.14±0.13	**72.00**±0.20	**70.33**±0.98	**47.98**±1.01	44.95±0.58
Upper Bound	97.22±0.22	97.78±0.22	96.91±0.03	97.44±0.03	77.07±0.47	81.67±0.18	75.59±0.61	87.71±0.45
Qwen3-32B (Non-thinking)								
Reasoning confidence	92.78±0.22	93.33±0.29	93.19±0.28	94.54±0.22	76.47±0.07	75.87±0.18	44.78±0.94	42.59±1.02
Answer confidence	92.56±0.11	92.22±0.29	93.84±0.05	93.42±0.13	75.40±0.46	74.67±0.18	**51.85**±0.61	44.11±0.94
Reasoning Confidence (normalised)	93.33±0.19	94.11±0.29	93.39±0.00	93.44±0.30	75.47±0.27	75.53±0.18	49.33±1.18	37.88±1.27
PiCSAR	93.22±0.22	93.55±0.33	93.90±0.28	93.88±0.22	**77.00**±0.18	75.93±0.13	46.91±1.02	**44.44**±2.28
PiCSAR-N	**93.33±0.38**	**93.89**±0.22	**94.12±0.03**	**94.09±0.26**	76.40±0.60	75.13±0.13	40.07±1.87	37.04±0.89
Upper Bound	96.78±0.11	98.00±0.00	96.28±0.13	96.99±0.07	82.27±0.13	83.73±0.07	72.56±1.87	86.20±1.02

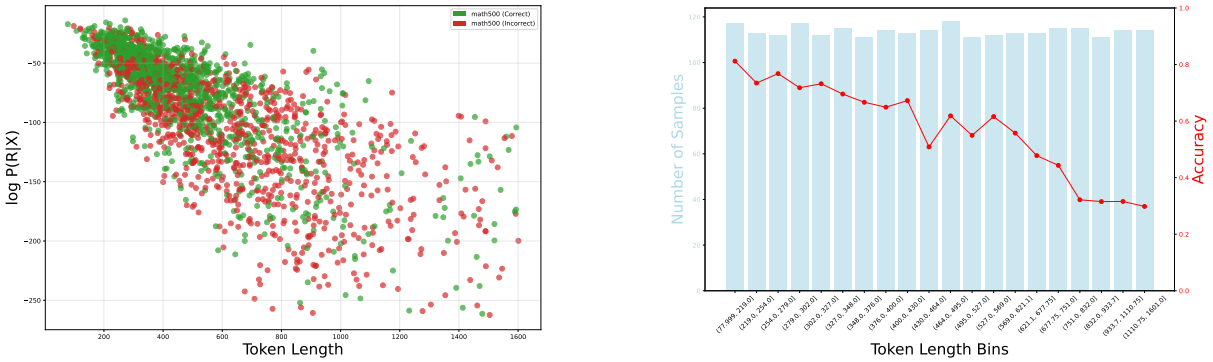
Table 7: **Performance comparison on benchmarks across methods on LLMs.** Values represent mean accuracy ± standard error over three independent evaluation runs. **Bold** indicates the best-performing method per column based on the mean accuracy. Sampling parameters: $k = \{6, 32\}$ for Gemma-2-9B, Llama-3.1-8B, and Qwen3-8B; $k = \{6, 16\}$ for Llama-3.1-70B and Qwen3-32B.

C.7 The Importance of Selection: Interpreting the Upper Bound

While PiCSAR consistently outperforms other heuristics, it necessarily falls short of the oracle *Upper Bound*, whose behaviour provides insight into the underlying challenges. On easier benchmarks such as SVAMP and GSM8K, the upper bound saturates quickly. For instance, increasing the sample size from $k = 6$ to $k = 32$ with Llama-3.1-70B on GSM8K raises accuracy only marginally from 96.91% to 97.44%, indicating that correct reasoning paths are usually present in small sample sets, and that selection rather than generation is the main bottleneck. In contrast, on more demanding tasks such as MATH500 and GPQA-Diamond, the upper bound continues to rise with larger k, as seen with Gemma-2-9B on GPQA-Diamond where accuracy jumps from 55.22% to 82.49%, reflecting the intrinsic difficulty of generating correct answers. In both regimes, PiCSAR demonstrates its value: in selection-limited settings, it reliably identifies correct candidates from small pools, while in generation-limited scenarios, it narrows the gap to the oracle by detecting correct reasoning even when correct answers are sparse, highlighting that improving selection is often as important as enlarging the sampling budget.

C.8 Analysis of Fallback Mechanism

To assess how sensitive our method is to the penalty assigned when a generation fails, *i.e.,* no answer token is produced and the answer-confidence term cannot be calculated, we tested several fallback values for the Answer Confidence score (Y). Specifically, we compared our default setting of $\log p(y \mid r, X) = -10$ with more conservative penalties of $Y = -20$ and $Y = -100$. As shown in Table 9, downstream accuracy is unchanged across all configurations. This indicates that, as long as the fallback value is sufficiently low to denote a failure state, its precise magnitude does not affect



(a) Length vs. Accuracy

(b) Information Plane Quadrants

Figure 7: Relationship between token length, probability, and accuracy.

Method	AIME 2024	AIME 2025	MATH500	SVAMP	GSM8K	GPQA-Diamond
			DS-Distill-llama-3-8B			
Reasoning Confidence	44.43±5.56	**35.56**±1.11	66.60±0.60	83.67±0.00	72.97±0.30	46.97±0.29
Reasoning Confidence (Normalised)	33.33±3.85	28.89±1.12	65.70±1.30	83.00±0.13	76.08±0.23	41.41±1.05
Answer Confidence	42.22±4.01	32.22±1.11	**67.60**±1.80	88.33±0.16	76.06±0.43	**48.99**±1.62
PiCSAR	**47.78**±4.01	33.33±1.13	67.20±0.60	85.67±0.07	**76.42**±0.16	47.31±0.17
PiCSAR-N	40.00±5.09	32.22±1.13	67.40±1.00	**89.00**±0.00	75.73±0.41	47.47±2.78
Upper Bound	66.67±5.09	51.11±1.11	82.00±0.13	95.67±0.00	92.91±0.35	77.27±0.77
			DS-Distill-Qwen-2.5-7B			
Reasoning Confidence	57.78±1.11	51.11±1.11	72.93±0.81	91.33±0.58	87.83±0.13	52.02±2.81
Reasoning Confidence (Normalised)	54.44±2.22	45.56±2.22	74.20±1.10	90.33±0.58	88.26±0.20	45.96±2.67
Answer Confidence	50.00±5.09	44.44±2.22	72.60±0.23	91.00±0.51	88.91±0.08	**53.20**±2.19
PiCSAR	**61.11**±1.11	**51.11**±1.11	**74.00**±0.70	**91.78**±0.48	88.18±0.07	52.36±2.88
PiCSAR-N	57.78±2.22	48.89±2.22	73.40±1.10	**91.78±0.29**	**89.60**±0.18	50.34±2.19
Upper Bound	72.22±1.11	70.00±0.00	83.33±0.18	96.33±0.38	96.79±0.13	79.12±2.07
			Qwen3-8B			
Reasoning Confidence	80.00±0.00	68.89±2.22	79.20±0.00	93.00±0.33	95.92±0.03	58.59±1.62
Reasoning Confidence (Normalised)	67.78±2.22	65.56±4.01	80.00±0.00	93.56±0.56	95.72±0.05	56.23±1.76
Answer Confidence	76.67±0.00	**73.33**±1.92	80.13±0.33	93.78±0.11	95.37±0.00	60.61±0.29
PiCSAR	**81.33**±1.34	68.89±2.22	80.60±0.13	**94.33**±0.33	**95.94**±0.04	59.43±1.61
PiCSAR-N	76.67±3.33	70.00±5.09	**89.67±0.37**	94.22±0.56	95.08±0.03	**61.11**±1.77
Upper Bound	87.78±1.11	82.22±1.11	84.00±0.12	97.56±0.11	97.54±0.03	80.13±0.45

Table 8: **Performance comparison of model across various baselines and benchmarks on LRMs, measured in terms of accuracy (%).** For all the evaluations, we use $k = 6$ sampling. *PiCSAR outperforms all baselines with more pronounced gains in more challenging benchmarks.*

candidate rankings.

$\log p(y \mid r, X)$	**Accuracy**
-10	53.40%
-20	53.40%
-100	53.40%

Table 9: Sensitivity analysis of the Answer Confidence fallback value (Y) on model accuracy. The performance is robust to the magnitude of the penalty.

C.9 Analysis of Performance with Number of Samples and Temperature

We first examine the scaling behavior of PiCSAR regarding the number of candidate generations (k). We evaluate GEMMA-2-9B on the SVAMP dataset with sample budgets ranging from $k = 6$ to $k = 32$. As shown in Table 10, PiCSAR exhibits scaling properties, with accuracy consistently improving as the candidate pool expands (rising from 89.11% at $k = 6$ to 90.22% at $k = 32$). In contrast, Self-Consistency plateaus earlier and is consistently out-

Samples	**PiCSAR (%)**	**Self-Consistency (%)**
6	89.11	88.15
10	89.89	88.56
16	89.89	88.11
32	90.22	88.89

Table 10: Scaling analysis of GEMMA-2-9B on SVAMP comparing PiCSAR against Self-Consistency across varying sample counts.

performed by PiCSAR. This indicates that PiCSAR is more effective at leveraging larger compute budgets to identify correct reasoning chains.

Additionally, we assess the stability of our method with respect to generation stochasticity by comparing performance at sampling temperatures of $T = 0.7$ and $T = 1.0$. The results, summarized in Table 10, reveal negligible performance variance (89.89% vs. 89.67%). These results indicate that PiCSAR is robust to moderate changes in generation hyperparameters and maintains high precision even under more stochastic sampling conditions ($T = 1.0$).

C.10 Nemenyi Post-hoc Test for PiCSAR

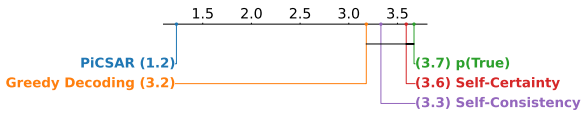


Figure 8: Critical difference diagram based on Nemenyi test ($p < 0.05$) for LLM. PiCSAR significantly outperforms all competing methods, while Self-Consistency, Self-Certainty, and p(True) show no significant difference from each other.

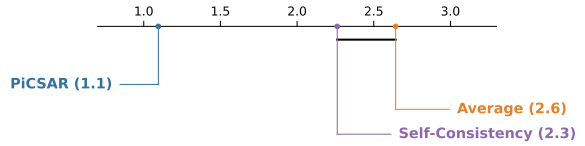


Figure 9: Critical difference diagram based on Nemenyi test ($p < 0.05$) for LRM. PiCSAR significantly outperforms the next-best methods, while Self-Consistency and Average show no significant difference from each other.

In Figure 8 and Figure 9 we show the critical diagrams obtained by performing the Nemenyi post-hoc test. In the critical diagram, the group of methods that do not differ significantly (significance level 0.05) are connected through a horizontal line.

At $p < 0.05$, the Nemenyi post-hoc test shows that PiCSAR (average rank ≈ 1.2) is significantly better than all other methods, as its rank is well separated beyond the critical difference from Greedy Decoding, Self-Consistency, Self-Certainty, and *p(True)*. This indicates that PiCSAR consistently outperforms the alternatives across datasets, and its superior performance is statistically robust rather than due to random variation.

As for LRM, statistical analysis utilising the Friedman test revealed highly significant performance differences across the methods ($p < 9.96 \times 10^{-7}$). Subsequent Nemenyi post-hoc comparisons confirmed that PiCSAR significantly outperforms both the Average and Self-Consistency baselines, showing mean rank differences of 1.55 and 1.17 respectively, both of which substantially exceed the critical difference of 0.72 at $\alpha = 0.05$.

D Sentence Level Analysis

In this section, we provide analysis between senetence level of PiCSAR and PiCSAR-N.

Our empirical analysis across AIME2025, AIME2024, and MATH500 with DEEPSEEK-R1-DISTILL-QWEN-2.5-7B demonstrates that this sensitivity is well-calibrated and reflects genuine quality differences rather than arbitrary length penalisation. Excessively long reasoning chains (>10K tokens) achieve only 14.1%, 13.5%, and 36.5% accuracy on these three benchmarks, compared to 92.7%, 82.1%, and 69.3% for the 1K–5K token range. Wrong answers are consistently 2–3× longer than correct ones (e.g., 15,761 vs. 5,651 tokens on AIME2025), confirming that excessive length may signal model uncertainty.

We further conducted a sentence-level trajectory analysis of $\log p(y|r, x)$ across reasoning chains for QWEN3-8B.

On AIME2024 (thinking mode):

- The highest-PICSAR generations (avg ∼650 sentences) begin with $\log p(y \mid r, x)$ around -9.5 at early sentence positions and rise steadily, peaking near -0.5 — demonstrating that each reasoning step productively advances toward the correct answer.

- The lowest-PICSAR generations (avg ∼1,700 sentences, ∼2× longer) have $\log p(y \mid r, x)$ oscillating erratically between -5 and -7 throughout, never reaching the sharp convergent peak observed in high-PICSAR samples.

On MATH500 (no-think mode), the same pattern holds:

- High-PICSAR generations (avg ∼660 sentences) show $\log p(y \mid r, x)$ climbing monotonically from -9.6 to a peak near -0.05.

- Low-PICSAR generations (avg ∼1,057 sentences, 1.6× longer) rise more slowly, plateau

Model	Metric	t	p-value	U-stat	d	Mean (C / I)
LLaMA-3.1-8B	$\log p(y \mid r, x)$	4.57	6.06×10^{-6}	38441	0.41	-4.21 / -5.75
	$\log p(r \mid x)$	9.11	2.00×10^{-18}	45115	0.82	-45.78 / -67.43
LLaMA-3.1-70B	$\log p(y \mid r, x)$	5.76	1.48×10^{-8}	41596	0.54	-0.41 / -1.47
	$\log p(r \mid x)$	6.99	8.76×10^{-12}	39096	0.66	-39.87 / -53.69
Gemma-2-9B	$\log p(y \mid r, x)$	9.03	3.70×10^{-18}	42086	0.81	-0.37 / -2.68
	$\log p(r \mid x)$	9.03	3.85×10^{-18}	45831	0.81	-18.64 / -30.80
Qwen3-8B	$\log p(y \mid r, x)$	5.37	1.24×10^{-7}	36835	0.54	-0.94 / -2.36
	$\log p(r \mid x)$	5.17	3.39×10^{-7}	31131	0.52	-41.88 / -68.41
Qwen3-32B	$\log p(y \mid r, x)$	6.09	2.26×10^{-9}	34500	0.64	-0.38 / -1.82
	$\log p(r \mid x)$	4.98	8.81×10^{-7}	27660	0.52	-61.92 / -95.85
Think-Qwen3-8B	$\log p(y \mid r, x)$	4.97	9.11×10^{-7}	27177	0.56	-2.17 / -4.55
	$\log p(r \mid x)$	2.67	0.008	21190	0.30	-418.77 / -587.10
Think-DS-R1 Distill-Qwen-7B	$\log p(y \mid r, x)$	3.87	1.21×10^{-4}	29105	0.39	-1.69 / -2.76
	$\log p(r \mid x)$	2.04	0.042	29023	0.21	-174.75 / -254.23
Think-DS-R1 Distill-LLaMA-8B	$\log p(y \mid r, x)$	5.99	4.00×10^{-9}	39822	0.57	-0.97 / -3.20
	$\log p(r \mid x)$	4.63	4.60×10^{-6}	31908	0.44	-246.18 / -500.00

Table 11: Statistical tests on Math500 comparing Correct (C) and Incorrect (I) samples. We report t-statistic (t), Mann-Whitney U (U), and Cohen's d (d) for both prediction confidence ($\log p(y \mid r, x)$) and reasoning compression ($\log p(r \mid x)$). All differences are significant at $p < 0.05$.

at substantially worse values (-12.0), and exhibit noisy fluctuations rather than clean convergence.

These trajectory analyses demonstrate that the additional length in low-PiCSAR generations does not yield meaningful progress in reasoning tasks — the model cycles through unproductive loops without converging. PiCSAR's unnormalised $\log p(r \mid x)$ faithfully captures this distinction by encoding the model's own confidence in a reasoning path, naturally assigning lower scores to long, uncertain chains. However, as shown in the example above regarding Qwen3-8B on AIME2024 and MATH500, when each generation converges to the answer, accuracy increases with the length. This length sensitivity is therefore a desirable property that rewards efficient, convergent reasoning over verbose, aimless generation.

We further provide three different cases in Appendix H: (1) a lengthy generation that arrives at the correct answer through extended thinking, and (2) a concise generation with high information density that likewise yields the correct answer. (3) The impact of answer confidence in generation selection.

E Additional Experiments for Confidence Information Plane

In this section, we show all the models across datasets (GSM8K, MATH500 and AIME2024), which consist of a variety of difficulties. We observe a consistent pattern across PiCSAR. In addition, the utility of our confidence metric extends to filtering for high-reliability answers. For GSM8K and MATH500, we use the median as our threshold with outliers removed, similar to §2.3. However, as for AIME2024, as the instance is similar, we include all the instances including the outliers, and set the threshold to 60% for both x and y-axis. We show results on GSM8K in Figure 10–14. Similarly, results on MATH500 are provided in Figures 15–19. We provide results on AIME 2024 in Figures 20–23. In addition, we also show results of using 75th percentile as the threshold in Figure 24. As shown in Figure 24, increasing the confidence thresholds from the median to the 75th percentile isolates a region in the Information Plane with significantly higher accuracy, effectively identifying the most trustworthy solutions.

E.1 Statistical Tests

In this part, we present detailed results of the statistical tests described in §2.3. We conduct these tests on the MATH500 dataset, with all results reported in Table 11.

The *terms* reported in the table correspond to reasoning confidence and answer confidence. The *t-statistic* measures the degree of separation between correct and incorrect responses. The associated *p-value* confirms that both confidence metrics significantly contribute to this distinction rather than

Method	Gemma-2-9B	Qwen3-8B	Llama-3.1-70B	DS-Qwen-7B	Average
Skywork-Reward-V2-Llama-3.1-8B	48.40	74.20	62.60	74.80	65.00
LMUnit-qwen2.5-72B	48.40	75.40	64.00	75.00	65.70
PiCSAR	46.53	73.67	63.67	74.00	64.47

Table 12: **Performance comparison of PiCSAR against trained verifiable reward models on the MATH500 dataset (%).** Despite being a zero-shot, training-free approach, PiCSAR achieves highly competitive performance compared to state-of-the-art reward models trained on large preference datasets.

Method	Gemma-2-9B	Qwen3-8B	Llama-3.1-70B	DS-Qwen-7B	Average
Skywork-Reward-V2-Llama-3.1-8B	89.60	93.17	89.83	89.37	90.49
LMUnit-qwen2.5-72B	88.84	93.55	94.53	89.37	91.57
PiCSAR	88.66	92.33	94.58	88.18	90.94

Table 13: **Performance comparison of PiCSAR against trained verifiable reward models on the GSM8K dataset (%).** Similar to the MATH500 results, the zero-shot, training-free PiCSAR demonstrates highly competitive performance against trained reward models.

arising from random variation. The *U-statistic* provides a non-parametric validation that the scores for correct responses are stochastically distinct from those for incorrect ones. The *Cohen's d* quantifies the magnitude of this effect size. The *mean (C/I)* indicates the directionality of the relationship, showing that the average log-probabilities are consistently higher for correct responses than for incorrect ones.

To observe whether the ability of the base model itself influence the performance of PiCSAR, we perform a one-way ANOVA testing whether the choice of base model affects PICSAR's improvement over the best competing baseline across all benchmarks. The result $F = 1.95$, $p = 0.15$ indicates no statistically significant difference, confirming that PICSAR provides consistent gains regardless of the underlying model. This is supported by our findings in 5.3, where we show that the answer-confidence component can be evaluated by a relatively smaller model without degrading selection quality, removing the requirement for the scoring model to match the generator in scale or architecture.

E.2 In Comparison between PiCSAR and Trained Verifiable Rewards

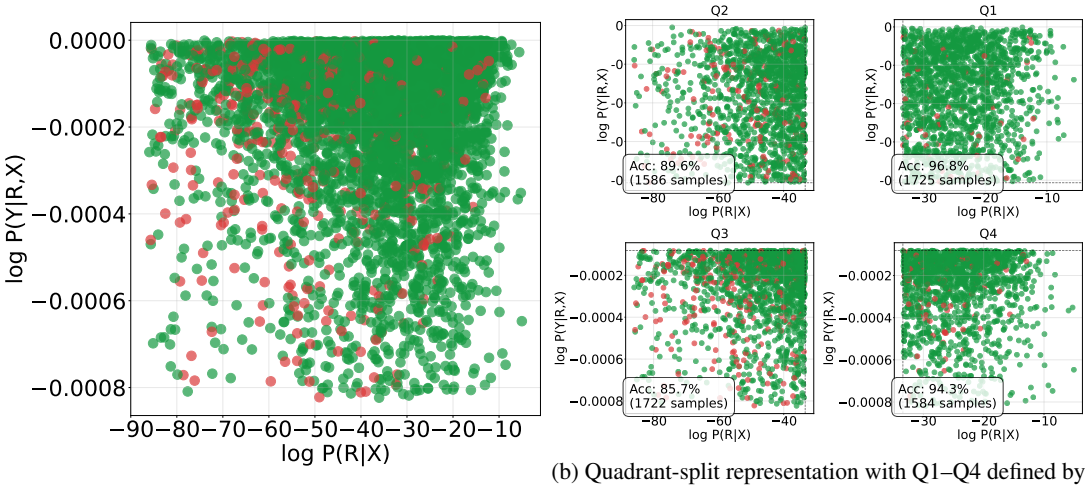
In this section, we compare PiCSAR with trained verifiable rewards. We evaluated PiCSAR with two best performing Reward Models from RewardBench (Lambert et al., 2025), *Skywork-Reward-V2-Llama-3.1-8B* and *LMUnit-qwen2.5-72B*.

Based on Table 12 and 13, PiCSAR achieves parity with (and occasionally outperforms) *Skywork-Reward-V2-Llama-3.1-8B*, a model explicitly trained on massive preference datasets, despite PiCSAR being zero-shot and training-free. While larger RMs (*i.e., LMUnit-qwen2.5-72B*) generally perform better (avg +1-2%), the fact that PiCSAR is competitive with trained verifiers confirms its high effectiveness, especially due to its zero additional training cost.

F Dataset Details

- SVAMP (Patel et al., 2021): https://github.com/arkilpatel/SVAMP, License: SVAMP License
- GSM8K (Cobbe et al., 2021): https://huggingface.co/datasets/openai/gsm8k, License: GSM8K License
- MATH (Hendrycks et al., 2021): https://huggingface.co/datasets/HuggingFaceH4/MATH-500, License: MATH License
- GPQA-Diamond (Rein et al., 2024): https://huggingface.co/datasets/Idavidrein/gpqa/viewer/gpqa_diamond/train, License: GPQA License
- TheoremQA (Chen et al., 2023a): https://huggingface.co/datasets/TIGER-Lab/TheoremQA, License: GPQA License
- AIME-2024: https://huggingface.co/datasets/Maxwell-Jia/AIME_2024, License: AIME-2024 License
- AIME-2025: https://huggingface.co/datasets/opencompass/AIME2025, License: AIME-2024 License

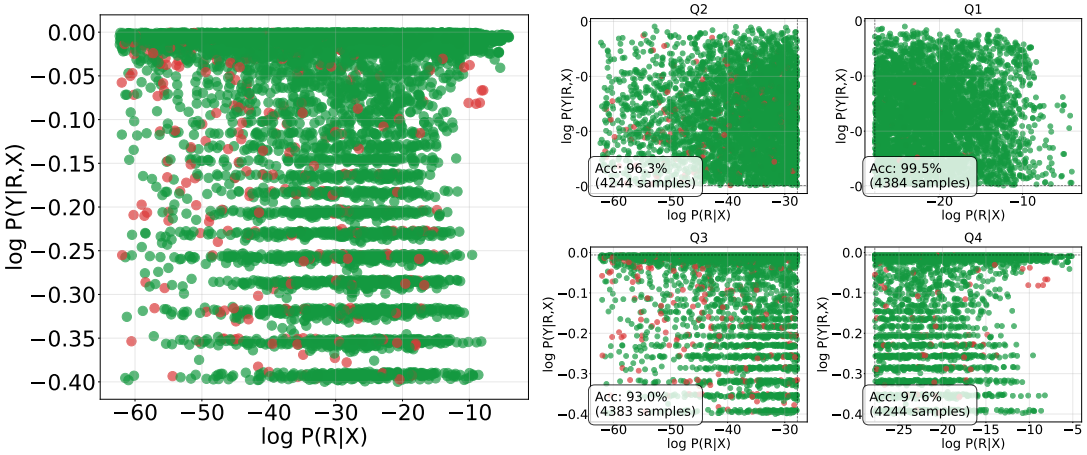
GSM8K



(a) Continuous distribution without quadrant partitioning.

(b) Quadrant-split representation with Q1–Q4 defined by median thresholds of reasoning and answer confidence.

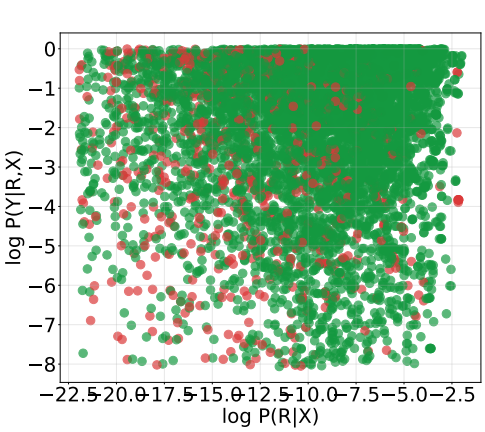
Figure 10: Information Plane visualisations of Llama-3.1-8B on the GSM8K dataset ($k = 6$). Green indicates correct answers, Red indicates incorrect ones.



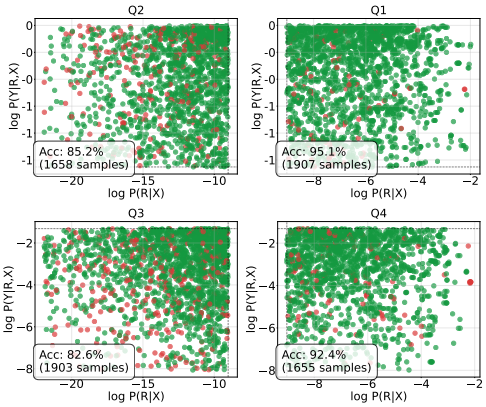
(a) Continuous distribution without quadrant partitioning.

(b) Quadrant-split representation with Q1–Q4 defined by median thresholds of reasoning and answer confidence.

Figure 11: Information Plane visualisations of Gemma-2-9B on the GSM8K dataset ($k = 6$). Green indicates correct answers, Red incorrect ones.

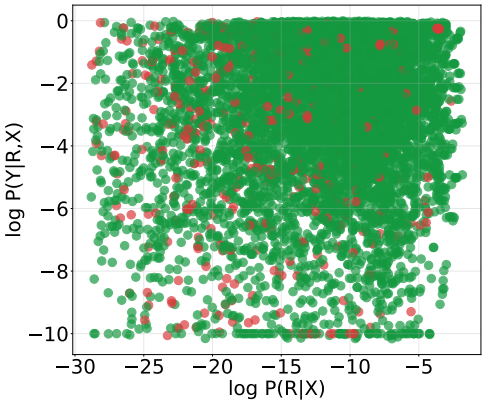


(a) Continuous distribution without quadrant partitioning.

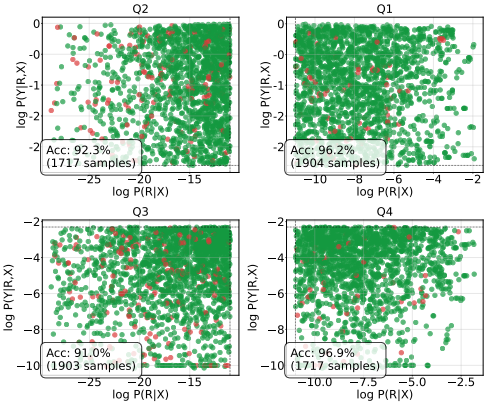


(b) Quadrant-split representation with Q1–Q4 defined by median thresholds of reasoning and answer confidence.

Figure 12: Information Plane visualisations of Gemma-2-9B on the GSM8K dataset ($k = 6$). Green indicates correct answers, Red incorrect ones.

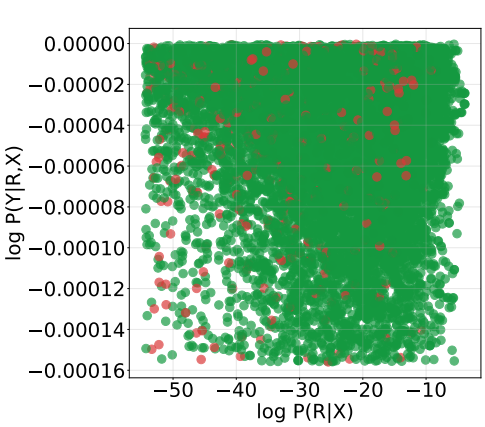


(a) Continuous distribution without quadrant partitioning.

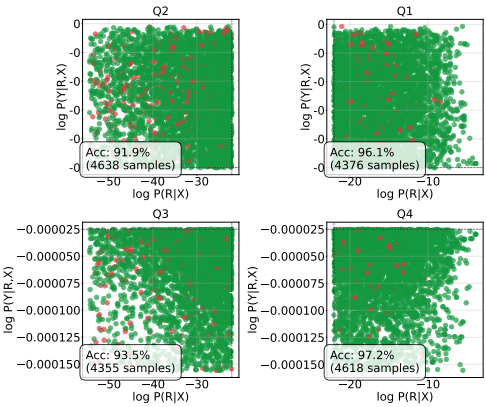


(b) Quadrant-split representation with Q1–Q4 defined by median thresholds of reasoning and answer confidence.

Figure 13: Information Plane visualisations of Gemma-2-9B on the GSM8K dataset ($k = 6$). Green indicates correct answers, Red incorrect ones.



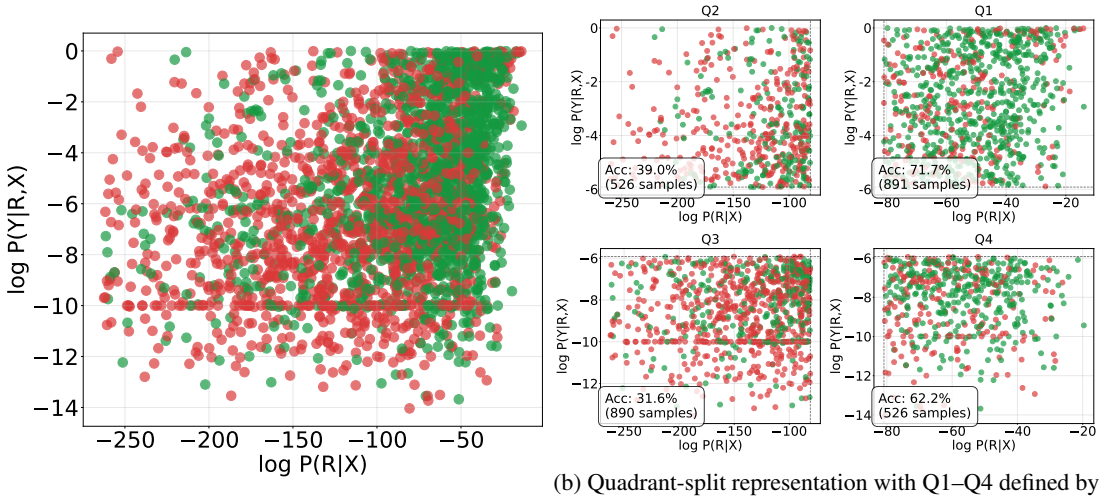
(a) Continuous distribution without quadrant partitioning.



(b) Quadrant-split representation with Q1–Q4 defined by median thresholds of reasoning and answer confidence.

Figure 14: Information Plane visualisations of Gemma-2-9B on the GSM8K dataset ($k = 6$). Green indicates correct answers, Red incorrect ones.

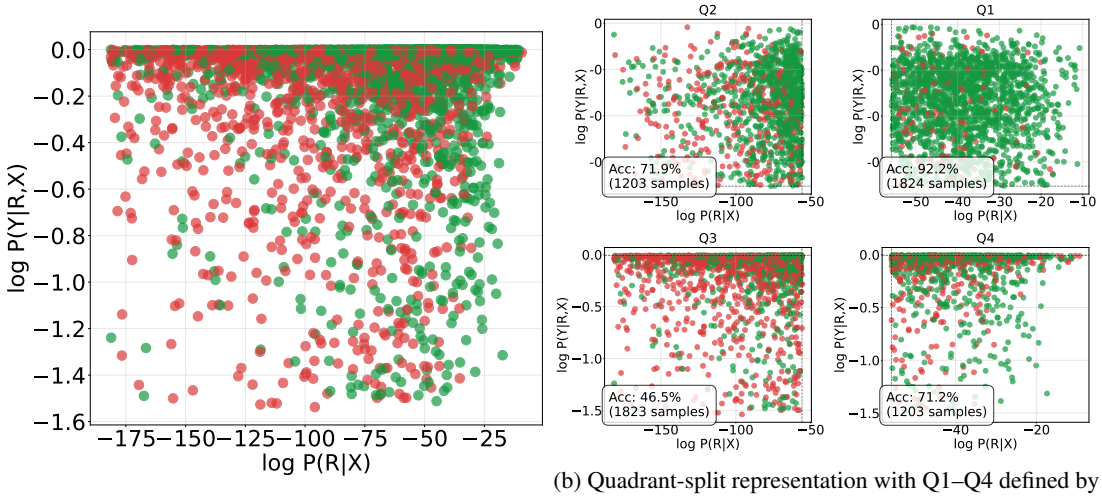
MATH500



(a) Continuous distribution without quadrant partitioning.

(b) Quadrant-split representation with Q1–Q4 defined by median thresholds of reasoning and answer confidence.

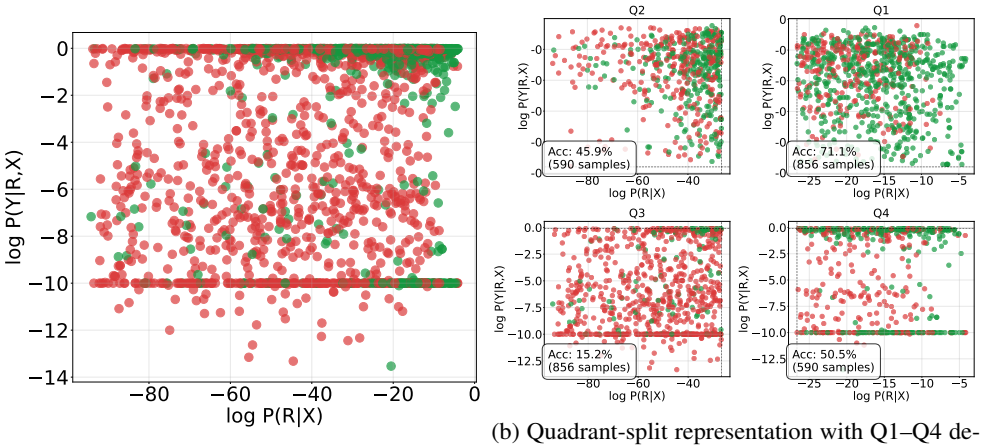
Figure 15: Information Plane visualisations of Llama-3.1-8B on the MATH500 dataset ($k = 6$). Green indicates correct answers, Red incorrect ones.



(a) Continuous distribution without quadrant partitioning.

(b) Quadrant-split representation with Q1–Q4 defined by median thresholds of reasoning and answer confidence.

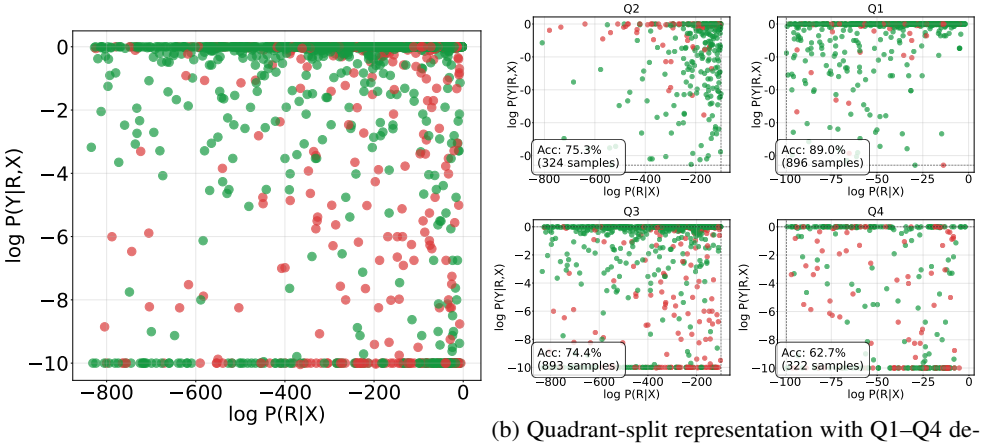
Figure 16: Information Plane visualisations of Llama-3.1-70B on the MATH500 dataset ($k = 6$). Green indicates correct answers, Red incorrect ones.



(a) Continuous distribution without quadrant partitioning.

(b) Quadrant-split representation with Q1–Q4 defined by median thresholds of reasoning and answer confidence.

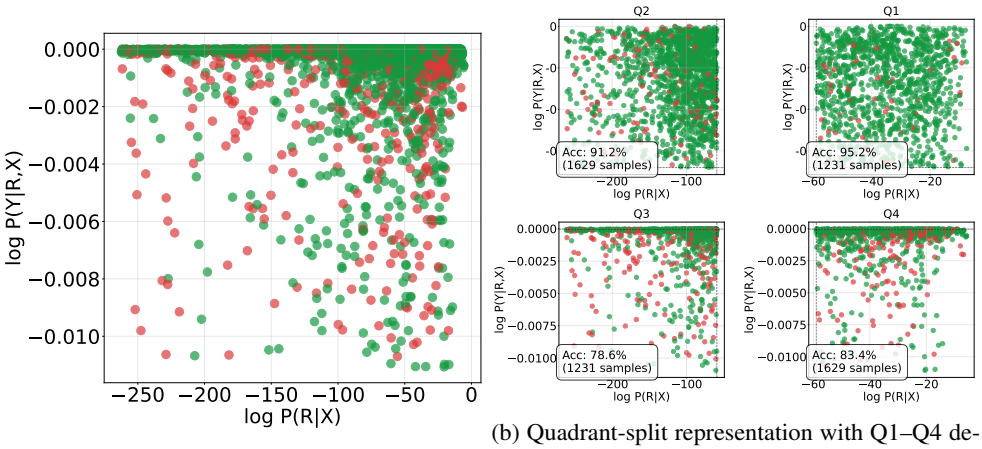
Figure 17: Information Plane visualisations of Gemma-2-9B on the MATH500 dataset ($k = 6$). Green indicates correct answers, Red indicates incorrect ones.



(a) Continuous distribution without quadrant partitioning.

(b) Quadrant-split representation with Q1–Q4 defined by median thresholds of reasoning and answer confidence.

Figure 18: Information Plane visualisations of Qwen3-8B on the MATH500 dataset ($k = 6$). Green indicates correct answers, Red incorrect ones.

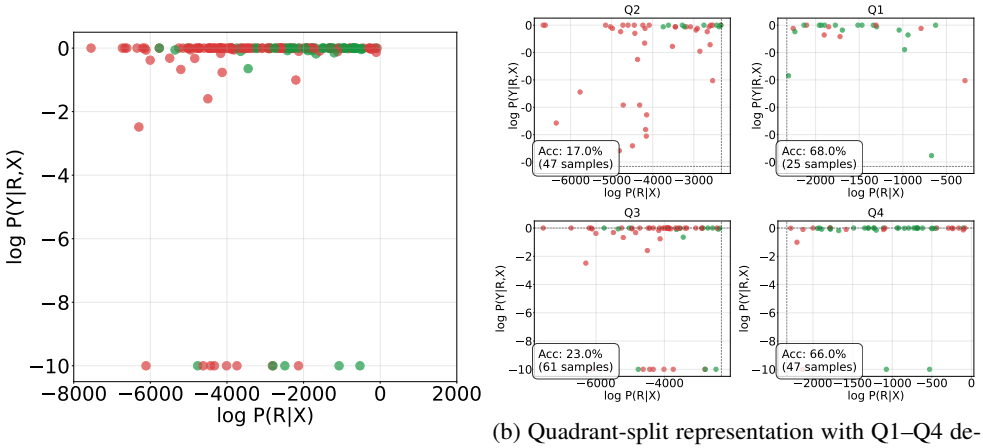


(a) Continuous distribution without quadrant partitioning.

(b) Quadrant-split representation with Q1–Q4 defined by median thresholds of reasoning and answer confidence.

Figure 19: Information Plane visualisations of Qwen3-8B on the MATH500 dataset ($k = 6$). Green indicates correct answers, Red incorrect ones.

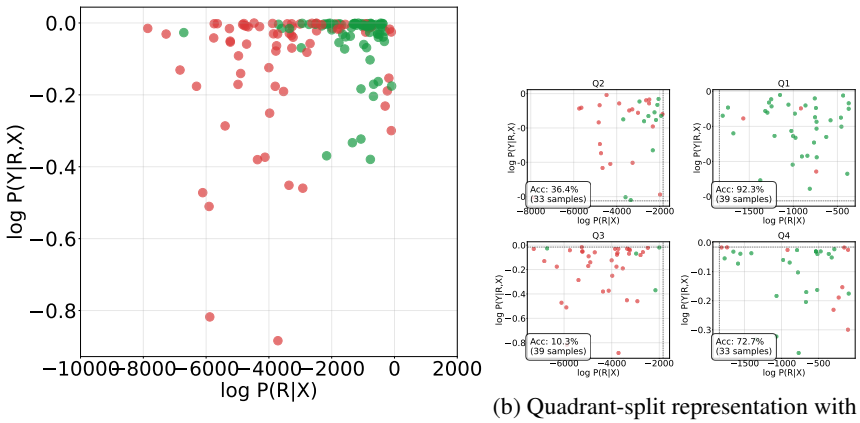
AIME2024



(a) Continuous distribution without quadrant partitioning.

(b) Quadrant-split representation with Q1–Q4 defined by median thresholds of reasoning and answer confidence.

Figure 20: Information Plane visualisations of DS-Distilled-Llama-8B on the AIME2024 dataset ($k = 6$). Green indicates correct answers, Red incorrect ones.



(a) Continuous distribution without quadrant partitioning.

(b) Quadrant-split representation with Q1–Q4 defined by median thresholds of reasoning and answer confidence.

Figure 21: Information Plane visualisations of DS-Distilled-Llama-8B on the AIME2024 dataset ($k = 6$). Green indicates correct answers, Red incorrect ones.

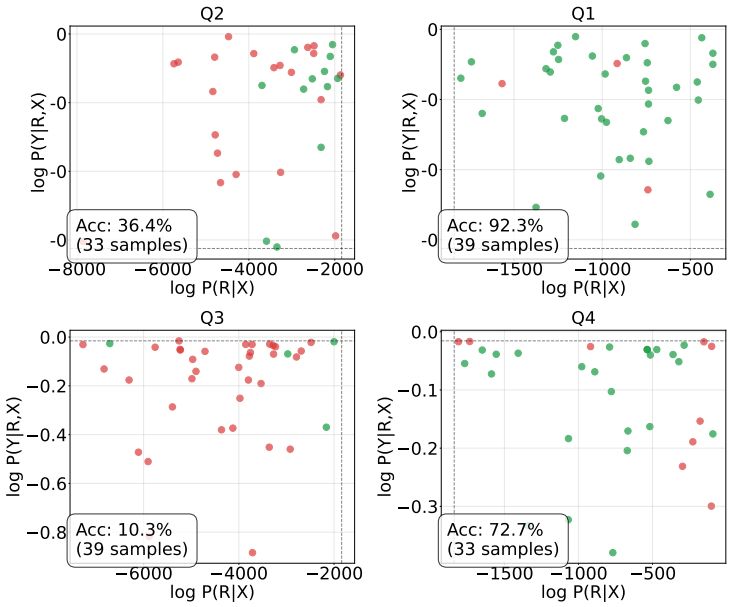
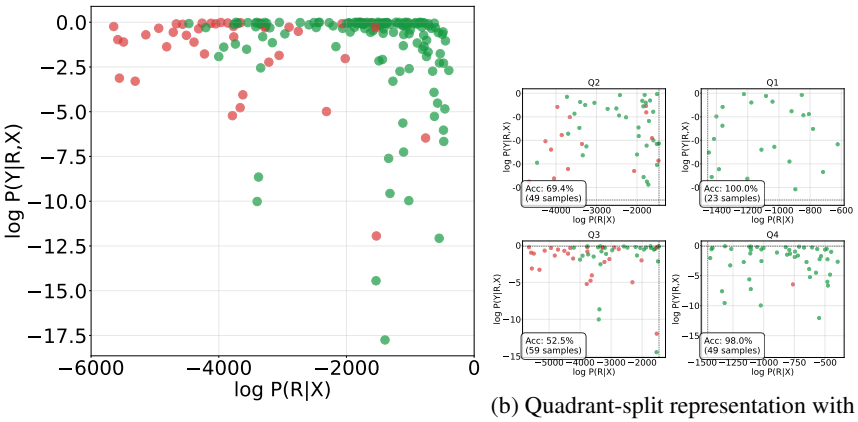


Figure 22: Information Plane of Deepseek-Distill-Qwen-2.5-7b on the AIME2024 dataset ($k = 6$). Dark red indicates correct answers; light red indicates incorrect ones. Quadrants represent: Q1 (high answer confidence, low reasoning confidence), Q2 (high both), Q3 (low both), and Q4 (high reasoning confidence, low answer confidence).

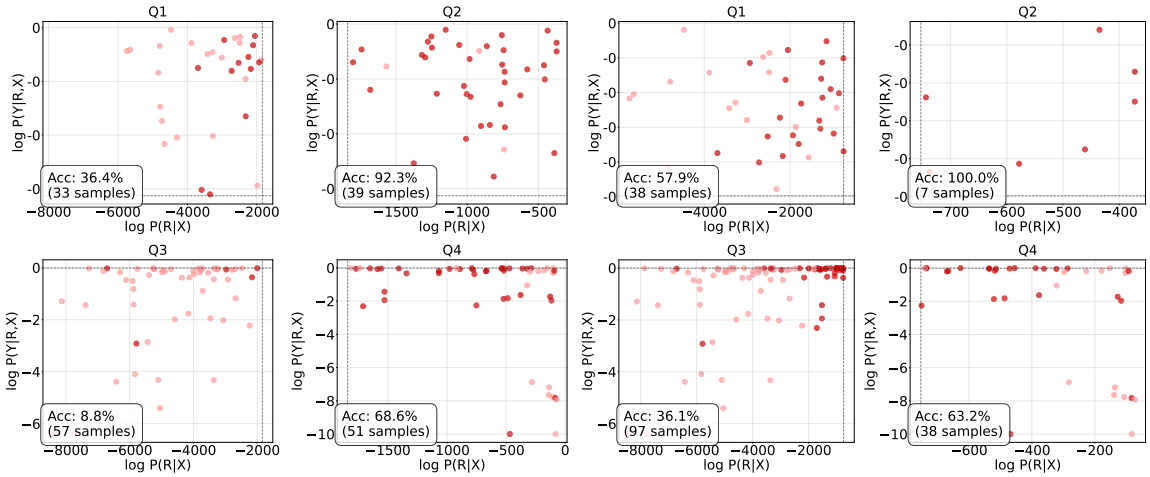


(a) Continuous distribution without quadrant partitioning.

(b) Quadrant-split representation with Q1–Q4 defined by median thresholds of reasoning and answer confidence.

Figure 23: Information Plane visualisations of DS-Distilled-Llama-8B on the AIME2024 dataset ($k = 6$). Green indicates correct answers, Red incorrect ones.

AIME2024 with various Quadrants



(a) Quadrants defined by the 50th percentile (median)

(b) Quadrants defined by the 75th percentile.

Figure 24: Effect of confidence thresholding on the Information Plane for DS-Distilled-Qwen-2.5-7B ($k = 6$) on AIME2024.

G Example of PiCSAR Scoring

Question: Every day, Wendi feeds each of her chickens three cups of mixed feed. She gives the flock three meals per day. In the morning, she gives 15 cups of feed, and in the afternoon she gives another 25 cups. How many cups of feed must she give in the final meal of the day if her flock contains 20 chickens?

High PiCSAR Score

```
Step 1: Determine the total number of cups of feed given in the morning. Wendi gives 15 cups.
Step 2: Determine the total number of cups of feed given in the afternoon. Wendi gives 25 cups.
Step 3: Total feed given so far = 15 cups + 25 cups = 40 cups.
Step 4: Each chicken needs 3 cups of feed per day.
Step 5: Total daily feed required = 3 cups/chicken * 20 chickens = 60 cups.
Step 6: Total given in morning and afternoon = 40 cups.
Step 7: Final meal feed = 60 cups - 40 cups = 20 cups.
The final answer is: 20.
PiCSAR Score: -36.76
```

Low PiCSAR Score

```
Step 1: Determine the total morning feed. Wendi gives 15 cups.
Step 2: Determine the total afternoon feed. Wendi gives 25 cups.
Step 3: Total feed in morning and afternoon = 15 + 25 = 40 cups.
Step 4: Since there are three meals, multiply the morning+afternoon total by 1.5 to estimate
total daily feed. Total = 40 * 1.5 = 60 cups.
Step 5: Each chicken is assumed to get 3 cups per meal, so 3 meals * 3 cups = 9 cups per chicken
per day.
Step 6: Total feed needed per day = 20 chickens * 9 cups = 180 cups.
Step 7: Final meal feed = 180 cups - 40 cups = 140 cups.
The final answer is: 140.
PiCSAR Score: -112.79
```

A concise generation with high token probability yields a higher PiCSAR score than a longer, lower-probability generation.

Figure 25: Example of PiCSAR scoring on a grade-school maths problem. The high-scoring generation is both correct and concise, while the low-scoring generation contains faulty reasoning and receives a substantially lower score.

H Further Case Study on of PiCSAR Scoring

H.1 Case 1 & 2

Case 1: Correct generation is shorter yet yields a higher log-probability

Question: There is one set of twins and one set of triplets. One twin is 7 years older than one triplet. If their combined ages are 44, how old is one of the twins?

Correct Generation (PiCSAR Score: -46.12)
Step 1: Let's denote the age of one of the twins as T and the age of one of the triplets as Tr. Since one twin is 7 years older than one triplet, we can express this relationship as T = Tr + 7.
Step 2: The combined ages of the twins and the triplets are 44. We can express this as 2T + 3Tr = 44.
Step 3: Substitute T = Tr + 7 into the equation: 2(Tr + 7) + 3Tr = 44.
Step 4: Expand: 2Tr + 14 + 3Tr = 44.
Step 5: Combine: 5Tr + 14 = 44.
Step 6: 5Tr = 30. Step 7: Tr = 6. Step 8: T = 6 + 7 = 13.
The final answer is: 13

Incorrect Generation (PiCSAR Score: -694.92)
Step 1: Let's denote the age of one triplet as 'x'. Combined age of triplets: 3x.
Step 2: Let's denote the age of one twin as 'y'. Combined age of twins: y + (y + 7) = 2y + 7.
Step 3: Equation: 3x + 2y + 7 = 44. Step 4: Simplify: 3x + 2y = 37.
Step 5: We can express 'y' in terms of 'x' from the equation y + (y + 7) = 2y + 7, which is y = (2y + 7) - 7. So, y = 2y. ...[TRUNCATED]... However, this is incorrect. Let's re-evaluate the equation y + (y + 7) = 2y + 7. We can represent it as y + 7 = y + 7. However, this is incorrect. Let's re-evaluate...

Case 2: Correct generation is longer yet yields a higher log-probability

Question: Milly needs to return a book she decided was really boring. The book weighs 4 pounds, cost $32, and needs to be returned to a distribution center 20 miles away. If the shipping company charges $0.35 per pound plus $0.08 per mile, and Amazon will only refund 75% of the book's purchase price, how much money will Milly lose?

Correct Generation (PiCSAR Score: higher by 105.90)
Step 1: Shipping cost by weight = 4 × $0.35 = $1.40.
Step 2: Shipping cost by distance = 20 × $0.08 = $1.60.
Step 3: Total shipping = $1.40 + $1.60 = $3.00.
Step 4: Refund = 75% × $32 = $24.
Step 5: Loss from refund = $32 - $24 = $8.
Step 6: Total loss = $3.00 + $8 = $11.
The final answer is: $11

Incorrect Generation
Step 1-2: Shipping = 4 × $0.35 + 20 × $0.08 ...[TRUNCATED]... The total loss is actually $2.99 - $24 = -$21.01. Since we cannot lose a negative amount, we must recalculate the total loss.
Step 6: Since the refund is more than the shipping cost, the total loss will be 0.
The final answer is: $2.99

In Case 1, the correct generation is shorter and scores 648.80 points higher. In Case 2, the correct generation is 67 tokens longer yet scores 105.90 points higher, demonstrating that PiCSAR rewards reasoning quality rather than brevity.

Figure 26: PiCSAR scoring examples illustrating that the score reflects reasoning quality independent of generation length.

H.2 Case 3

Case 3: Impact of Answer Confidence

Question: In the first week, Judy read for 15 minutes each night before going to sleep. In the second week, she read a total of 100 pages. If she can read 2 pages per 1.5 minutes, how many pages did she read in two weeks?

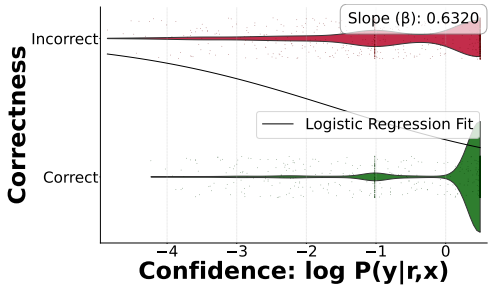
Before Answer Confidence (Incorrect Generation)
Step 1: Total reading time in the first week: 15 × 7 = 105 minutes.
Step 2: Pages read in the first week: 105 / 1.5 = 70 pages.
Step 3: Total pages in two weeks: 70 + 100 = 170 pages.
The final answer is: 170

After Answer Confidence (Correct Generation)
Step 1: Total reading time in the first week: 15 × 7 = 105 minutes.
Step 2: Pages read in the first week: 105 / 1.5 × 2 = 140 pages.
Step 3: Total pages in two weeks: 140 + 100 = 240 pages.
The final answer is: 240

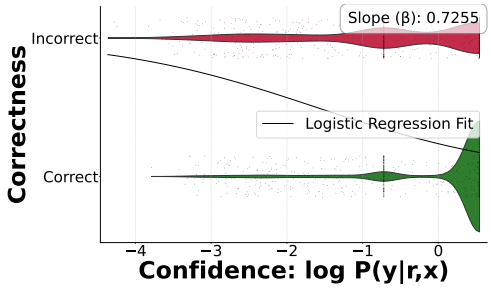
Correct Answer: 240

Both generations share similar reasoning structure and length. The incorrect generation omits the multiplication by 2 in Step 2, yielding a lower answer confidence that allows PiCSAR to discriminate between the two.

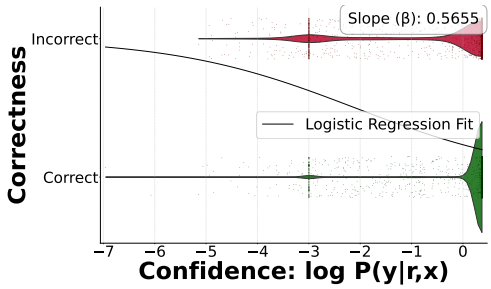
Figure 27: Example illustrating the role of answer confidence in PiCSAR scoring.



(a) Qwen3 8B



(b) Qwen3 14B



(c) Qwen3 32B

Figure 28: A detailed visualisation on the correct/incorrect densities based on logistic regression plot.

I Intra-model Reliability

To support the intra-model results in Section 5.2, we analyse the calibration of PiCSAR's confidence signal using the evaluation traces collected for the Qwen3 family. For every sample we pair the answer log-probability $\log p(y \mid r, x)$ with its correctness label and fit a separate model per backbone. The resulting calibration curves in Figure 4 exhibit a consistent monotonic trend: the logistic slopes are 0.63, 0.73, and 0.57 for Qwen3-8B, 14B, and 32B respectively, and the corresponding point-biserial coefficients ($r \approx 0.31, 0.35, 0.29$) show a positive correlation between higher confidence and the probability of a correct answer.

Figure 28 also shows how this effect manifests in the raw score distribution. Correct solutions concentrate around higher confidence values (closer to zero log-probability), whereas incorrect ones remain several nats lower, leaving limited overlap in the high-confidence region.

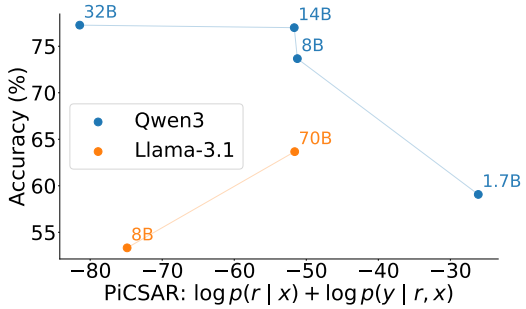


Figure 29: Comparison of % and PiCSAR score.

I.1 Logistic Regression Experimental Training

We model the relationship between confidence and correctness using logistic regression, similar to Gema et al. (2025b). The binary outcome variable encodes whether the final answer is correct ($y \in 0, 1$), while the predictor is the model's confidence score expressed as the log-probability of the final answer:

$$Pr(y = 1 \mid \mathit{Conf}) = \sigma(\alpha + \beta \cdot \mathit{Conf})$$

where σ is the sigmoid function. The regression coefficient β quantifies the change in log-odds of correctness per unit change in confidence. A positive β indicates that higher confidence increases the likelihood of correctness. For instance, as shown in Figure 28b, in Qwen3-14B, $\beta = 0.7255$ corresponds to more than doubling the odds of correctness ($e^{0.7255} \approx 2.07$).

I.2 Point-biserial Correlation Coefficient

As a complementary measure to logistic regression, we compute the point-biserial correlation coefficient between confidence scores (continuous) and correctness (binary). This statistic, mathematically equivalent to Pearson's correlation with a dichotomous variable, directly quantifies the strength of association between the two. It is defined as

$$r = \frac{\bar{x}_1 - \bar{x}_0}{s_x} \sqrt{\frac{n_1 n_0}{n^2}},$$

where $\bar{x}_1$ and $\bar{x}_0$ denote the mean confidence scores for correct and incorrect samples, s_x is the pooled standard deviation, and n_1, n_0 are the respective sample counts. The coefficient is bounded in $[-1, 1]$, with positive values indicating alignment between confidence and correctness. For instance, an r of 0.35 for Qwen3-14B indicates a moderate

positive association. Together with logistic regression, this provides a scale-free validation that confidence is a consistent predictor of correctness within a given model.

J Inter-model Variance

Inter-model variance analysis challenges the assumption that confidence scores represent universal correctness measures across different models. While intra-model reliability remains stable across different model sizes and architectures, confidence scores cannot be compared across models of different parameter sizes and architectures. As shown in Figure 29, the Llama family exhibits predictable trend: both accuracy and confidence increase with model size. In contrast, the Qwen family shows a non-monotonic relationship; Qwen3-1.7B achieves the highest confidence while showing the lowest accuracy. *This difference implies that while there is a general trend that confidence is a useful proxy for selecting an accurate reasoning path from a set of candidates within models, but its actual value is model-specific and incomparable across different models.*