

# Chronos: Learning Temporal Dynamics of Reasoning Chains for Test-Time Scaling

Kai Zhang\*, Jiayi Liao\*, Chengpeng Li, Ziyuan Xie, Sihang Li<sup>†</sup>, Xiang Wang<sup>†</sup>

University of Science and Technology of China  
kaizhang99@mail.ustc.edu.cn,  
{joyliao7777, sihang0520, xiangwang1223}@gmail.com

## Abstract

Test-Time Scaling (TTS) has emerged as an effective paradigm for improving the reasoning performance of large language models (LLMs). However, existing methods — most notably majority voting and heuristic token-level scoring — treat reasoning traces or tokens equally, thereby being susceptible to substantial variations in trajectory quality and localized logical failures. In this work, we introduce **Chronos**, a lightweight and plug-and-play chronological reasoning scorer that models each trajectory as a time series. Specifically, Chronos learns to capture trajectory features of token probabilities, assigns quality scores accordingly, and employs a weighted voting mechanism. Extensive evaluations on both in-domain and out-of-domain benchmarks demonstrate that Chronos consistently delivers substantial gains across a variety of models, with negligible computational overhead. Notably, Chronos@128 achieves relative improvements of 34.21% over Pass@1 and 22.70% over Maj@128 on HMMT25 using Qwen3-4B-Thinking-2507, highlighting its effectiveness. Our code is released on <https://github.com/Hizkai/Chronos>.

## 1 Introduction

Test-Time Scaling (TTS) (Wu et al., 2025; Brown et al., 2024) has emerged as a powerful paradigm that complements model training, substantially improving the performance of large language models (LLMs) (OpenAI, 2025; Yang et al., 2025; Gemini-Team, 2025; DeepSeek-AI, 2024) across a wide range of complex reasoning tasks (Kamalloo et al., 2023; Li et al., 2025b; Chen et al., 2025). A key driver of these gains is the use of parallel multi-sample aggregation strategies, which generate multiple independent reasoning trajectories and combine them to infer a consensus solution.

\*Equal Contribution.

<sup>†</sup>Corresponding Author.

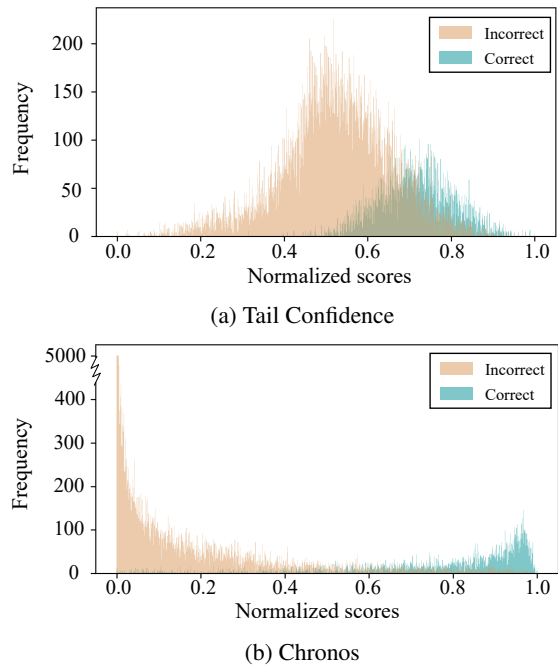


Figure 1: Distribution of scores for correct and incorrect trajectories on AIME25. (a) Tail Confidence and (b) Chronos. All experiments use the DeepSeek-1.5B with 128 samples for 16 repeats per question.

Early self-consistency methods aggregate final answers via majority voting (Wang et al., 2023). While effective, this approach suffers from a fundamental limitation: it treats all reasoning trajectories as equally reliable, ignoring substantial variations in their quality. As a result, erroneous or low-quality traces can disproportionately influence the aggregation, leading to suboptimal predictions. This limitation raises a critical research question: how can the quality of reasoning trajectories be accurately estimated?

Recent work (Fadeeva et al., 2024; Kang et al., 2025; Fu et al., 2025) has leveraged token-level distributional statistics to assess the quality of reasoning trajectories. However, these methods largely depend on predefined heuristics — such as token-level uncertainty (Fadeeva et al., 2024) or confidence scores (Fu et al., 2025) — and aggregate

them into a single trace quality estimate via uniform mean pooling. This design implicitly assumes that all tokens are unordered and contribute equally to the logical validity of a reasoning trajectory. In practice, such homogenization can mask critical failures at intermediate steps, impairing the detection of localized reasoning errors and substantially weakening discriminative power. As shown in 1, although tail confidence scores (Fu et al., 2025) can distinguish between correct and incorrect reasoning trajectories, there are still significant overlaps that incur ambiguity, resulting in suboptimal estimation of trajectory quality.

To overcome the limitations aforementioned, we introduce **Chronos** (**Chronological Reasoning Scorer**), a versatile, plug-and-play module for high-fidelity trajectory aggregation. In contrast to heuristic scorers (Fu et al., 2025) that model reasoning traces as unordered collections of token-level statistics, we posit that the validity of a reasoning chain is inherently a chronological process. Chronos maps each token to a discrete timestamp and explicitly models the directional progression of reasoning, enabling the scorer to capture sequential dependencies. We implement Chronos as a lightweight learned temporal scoring model that treats the inference trajectory as a time series. The model ingests sequences of token-level probabilities — reflecting the model’s internal signals — and produces a scalar quality score for each reasoning trajectory. Specifically, Chronos employs a multi-scale convolutional architecture to process the internal temporal signal. By utilizing parallel convolutional filters with varying kernel lengths, the model simultaneously captures local fluctuations and global dependencies within the reasoning process. These multi-scale blocks are stacked within a deep residual framework to capture complex temporal patterns, enabling high-fidelity estimation of trajectory quality.

We evaluate Chronos by training it on AIME (2000–2023) and conducting both in-domain and out-of-domain evaluations across multiple reasoning benchmarks — AIME25, HMMT25, and GPQA-Diamond — and model scales, including DeepSeek-1.5B, Qwen3-4B, and DeepSeek-8B. Across all settings, Chronos consistently yields substantial gains in TTS performance while introducing negligible computational overhead. Notably, when paired with Qwen3-4B-Thinking, Chronos@128 achieves 74.38% accuracy on HMMT25 — effectively saturating the

benchmark — compared to 55.42% for Pass@1 and 60.62% for Maj@128 (majority voting), with only a 0.0005% increase in inference FLOPs.

## 2 Related Work

**Scaling Test-Time Compute.** Recent advancements in LLMs (Wu et al., 2025; Brown et al., 2024; Snell et al., 2025) have emphasized scaling test-time compute along two complementary paradigms: sequential refinement and parallel exploration. Sequential scaling aims to deepen reasoning by iteratively refining solutions (Madaan et al., 2023) or by encouraging longer Chain-of-Thought (CoT) (Wei et al., 2022) traces through reinforcement learning (Jaech et al., 2024; DeepSeek-AI, 2024; Yang et al., 2025). In contrast, parallel scaling broadens the search space via multi-trajectory sampling (Wang et al., 2023; Li et al., 2025a) or structured tree search methods (Yao et al., 2023; Xie et al., 2024). Despite their effectiveness, a key bottleneck lies in synthesizing the parallel outputs. Existing approaches largely rely on predefined heuristics (Huang et al., 2024; Knappe et al., 2024; Kang et al., 2025; Fu et al., 2025), which struggle to distinguish correct solutions from hallucinations, as they either treat all reasoning steps uniformly or apply rule-based aggregation (*e.g.*, uniform mean pooling), disregarding the temporal dynamics of the reasoning process. To address this limitation, we model reasoning as a chronological process and leverages internal signals to accurately estimate trajectory quality.

**Time Series Modeling.** Modeling sequential data is central to various domains, evolving significantly from traditional distance-based heuristics (Berndt and Clifford, 1994; Lin et al., 2003) to modern deep learning approaches. To capture complex non-linear dependencies and long-term correlations, researchers have explored diverse architectures, including MLP-based mixers (Oreshkin et al., 2020; Zeng et al., 2023), recurrent networks (Lai et al., 2018; Flunkert et al., 2017), temporal convolutional networks (TCNs) (Bai et al., 2018; Liu et al., 2022), and Transformers (Li et al., 2019; Zhou et al., 2021). Drawing inspiration from this progress, we reframe the evaluation of LLM reasoning chains as a time-series classification task. We treat token probability sequences as temporal signals and propose a lightweight aggregator based on the InceptionTime architecture (Szegedy et al., 2015; Fawaz et al., 2020). By leveraging multi-

scale convolutional filters, our method effectively captures the chronological evolution of the reasoning process, achieving high-fidelity trajectory scoring with negligible computational overhead.

### 3 Methodology

In this section, we introduce **Chronos**, a lightweight and plug-and-play chronological reasoning scorer that models reasoning trajectories as temporal processes (*cf.* Figure 2). We first describe the token-level statistics that serve as the temporal signals for Chronos in Section 3.1. We then present the Chronos architecture in Section 3.2, detailing how it captures sequential dependencies in the reasoning process. Finally, in Section 3.3, we introduce the weighted majority voting scheme that leverages the predicted scores for aggregation.

#### 3.1 Token-level Statistics

Transformer-based (Vaswani et al., 2017) LLMs generate output token sequences  $y = (y_1, \dots, y_m)$  autoregressively, conditioned on an input sequence  $x = (x_1, \dots, x_n)$ . At each decoding step  $t$ , the model maps the preceding context to a logit vector  $\mathbf{v}_t \in \mathbb{R}^{|\mathcal{V}|}$ , where  $\mathcal{V}$  denotes the vocabulary. These logits are transformed via the softmax operator into a probability distribution over the next token:  $y_t$ :

$$P_t(\cdot | x, y_{<t}) = \text{Softmax}(\mathbf{v}_t) \in [0, 1]^{|\mathcal{V}|}. \quad (1)$$

This token-level distribution reflects the model’s confidence in predicting the next token.

Unlike prior approaches that collapse token-level signals into a single scalar via uniform pooling, Chronos preserves fine-grained sequential dependencies by retaining token-level probabilities of the reasoning trajectory as a chronological sequence  $\mathbf{s}$ :

$$\mathbf{s} = (s_1, s_2, \dots, s_L), \quad (2)$$

$$s_t = -\frac{1}{k} \sum_{i=1}^k \log P_t(i | x, y_{<t}) \quad (3)$$

where  $s_t$  denotes the negative mean log-probability of the top- $k$  candidate tokens at decoding step  $t$ . High values of  $s_t$  corresponds to more peaked distributions and greater model confidence, while low values indicate increased uncertainty. This chronological representation preserves the temporal structure of the reasoning process, enabling Chronos to capture localized confidence fluctuations that are obscured by uniform pooling strategies.

### 3.2 Model Architecture

Inspired by prior work in temporal signal processing (Szegedy et al., 2015; Fawaz et al., 2020), which shows that lightweight convolutional architectures can effectively capture temporal patterns, Chronos models the dynamic evolution of token-level statistics throughout the reasoning process, enabling accurate estimation of trajectory quality.

#### 3.2.1 Multi-scale Feature Extraction

The core is the multi-scale convolutional block, designed to process the temporal signal  $\mathbf{s}$  at varying windows. Motivated by the observation that the critical reasoning steps and final answer are typically concentrated in the last stages of the trajectory, Chronos focuses exclusively on the final  $L_{tail}$  tokens. Formally, we define the input as the temporal sequence  $\mathbf{s} \in \mathbb{R}^{1 \times L_{tail}}$ , where the channel dimension is 1 and the sequence length is  $L_{tail}$ . To augment model capacity, we first apply  $N_{Proj}$  filters with the shape of  $1 \times 1$  to project it into  $\mathbf{z}$ :

$$\mathbf{z} = \text{Conv}(\mathbf{s}, 1, N_{Proj}) \in \mathbb{R}^{N_{Proj} \times L_{tail}}, \quad (4)$$

where  $\text{Conv}(\cdot)$  represents the convolution operation utilizing  $N_{Proj}$  filters with the shape of  $1 \times 1$ . The expanded signal is then processed by parallel convolutional filters with varying kernel lengths  $l$ :

$$\mathbf{h}_l = \text{ReLU}(\text{Conv}(\mathbf{z}, l, N_{Conv})) \in \mathbb{R}^{N_{Conv} \times L_{tail}}. \quad (5)$$

This process involves the convolution operation utilizing  $N_{Conv}$  filters with the shape of  $1 \times l$ . Particularly, shorter filters are sensitive to local fluctuations and immediate inconsistencies in the reasoning steps, while longer filters capture broader trends and long-term dependencies within the reasoning trajectory. The outputs of these parallel convolutions, including  $\mathbf{z}$  — spanning local details and global context — are concatenated along the channel dimension to form a rich, multi-scale representation  $\mathbf{o} \in \mathbb{R}^{(N_{Proj} + k \cdot N_{Conv}) \times L_{tail}}$ :

$$\mathbf{o} = \text{Concat}(\mathbf{z}, \mathbf{h}_{l_1}, \mathbf{h}_{l_2}, \dots, \mathbf{h}_{l_k}), \quad (6)$$

where  $k$  is the number of distinct kernel lengths, and  $\mathbf{h}_{l_i}$  denotes the output feature map generated by the convolution with kernel length  $l_i$ .

#### 3.2.2 Deep Residual Architecture

To facilitate the training of a deep network capable of capturing long-context temporal patterns, we stack  $N_{Blk}$  multi-scale convolutional blocks in a

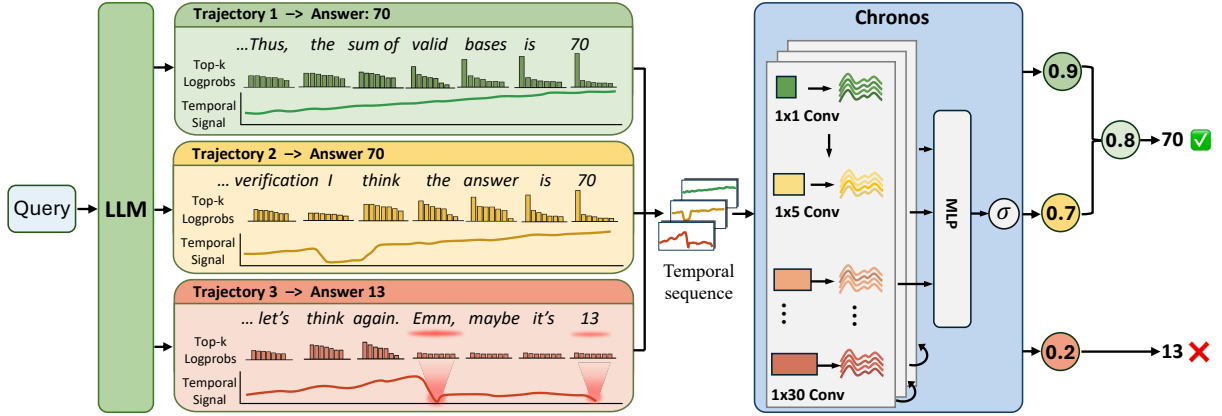


Figure 2: Framework of **Chronos**. It consists of three stages: (1) **Multi-trajectory Sampling**: Given an input query, we sample multiple independent inference trajectories and extract their token-level probability distributions, which are treated as temporal signals. (2) **Chronological Reasoning Scoring**: Chronos processes these temporal signals using multi-scale convolutions to explicitly capture sequential dependencies, producing a scalar quality score for each trajectory. (3) **Weighted Majority Voting**: The predicted trajectory scores are used to weight candidate answers, which are then aggregated to determine the final output.

sequential manner, integrated with residual connections for each block to mitigate the vanishing gradient problem. Formally, let  $\mathbf{M}_i(\cdot)$  denote the  $i$ -th convolutional block and  $\mathbf{o}_i$  represent the output of  $\mathbf{M}_i$ . The forward process is defined as:

$$\mathbf{o}_i = \mathbf{M}_i(\mathbf{o}_{i-1}), \hat{y} = \sigma(\text{MLP}(\sum_{i=0}^{N_{Blk}} \mathbf{o}_i)), \quad (7)$$

where  $\mathbf{o}_0 = \mathbf{z}$  is the expanded input embedding,  $\sigma(\cdot)$  is the sigmoid function, and  $\hat{y}$  denotes the final predicted score. For a given trajectory  $\tau_i$  from dataset  $\{\tau_i\}_{i=1}^N$ , we assign a ground-truth label  $y_i \in \{0, 1\}$  based on the correctness of the final answer, where  $y_i = 1$  indicates a correct prediction and  $y_i = 0$  denotes an error. The model is trained to minimize the Binary Cross-Entropy (BCE) loss between the predicted score  $\hat{y}_i$  and the label  $y_i$ :

$$\mathcal{L} = - \sum_{i=1}^N [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)]. \quad (8)$$

In practice, an ensemble strategy is adopted wherein multiple well-trained models independently evaluate the trajectory, with their average serving as the final score. As shown in Figure 1, by modeling reasoning traces as time series, Chronos effectively reduce the overlapping ambiguity among trajectories and substantially improves the accuracy of trajectory quality prediction.

### 3.3 Score-Weighted Majority Voting

To further boost aggregation performance, we implement a score-based filtering (Fu et al., 2025)

alongside weighted majority voting. Formally, given a question  $Q$ , we sample  $N_{smp}$  reasoning trajectories  $\{\tau_i\}_{i=1}^{N_{smp}}$  and predict their corresponding scores  $\{s_i\}_{i=1}^{N_{smp}}$ . We sort the trajectories in descending order of  $s_i$  and identify the subset of indices  $\mathcal{I}_{top}$ :

$$\mathcal{I}_{top} = \{i \mid \text{rank}(\hat{y}_i) \leq \lfloor \eta \cdot N_{sam} \rfloor\}, \quad (9)$$

where  $\eta$  is the retention ratio. The final answer  $\hat{a}$  is then determined by a weighted majority vote over this filtered subset:

$$\hat{a} = \arg \max_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}_{top}} \hat{y}_i \cdot \mathbb{I}(a_i = a), \quad (10)$$

where  $\mathcal{A}$  denotes the set of unique candidate answers for  $Q$ ,  $a_i$  represents the final answer extracted from trajectory  $\tau_i$ , and  $\mathbb{I}(\cdot)$  is the indicator function, which equals 1 if  $a_i$  matches candidate  $a$ , and 0 otherwise. This approach prioritizes high-quality reasoning traces by retaining only the top  $\eta$  of trajectories based on their predicted scores, thereby ensuring the final consensus relies exclusively on the most reliable traces.

## 4 Experiments

We conducted extensive experiments to demonstrate the effectiveness of Chronos. Specifically, our experiments aim to address the following research questions:

- **RQ1**: How does Chronos perform compared to existing TTS methods?

- **RQ2:** What is the computational overhead introduced by Chronos, and how does accuracy evolve with the scaling of test-time computation?
- **RQ3:** How well does Chronos generalize when applied to different sampling models?

#### 4.1 Experimental Setup

**Models.** We evaluate Chronos across multiple model scales, including: **DeepSeek-1.5B**<sup>1</sup>, **Qwen3-4B**<sup>2</sup> and **DeepSeek-8B**<sup>3</sup>. We selected these models due to their proven efficacy in reasoning tasks. Complete generation settings and prompting templates are provided in Appendix A.

**Datasets.** To prevent data leakage, we train Chronos exclusively on the AIME archive spanning 2000 – 2023, ensuring that the training data is comparable in difficulty to the evaluation benchmarks. We evaluate Chronos on AIME25, HMMT25 (Feb), and GPQA-Diamond, which are widely used benchmarks for assessing the reasoning capabilities of frontier LLMs. Notably, AIME and HMMT consist of competitive mathematics problems, whereas GPQA-Diamond comprises scientific reasoning questions, thereby enabling a rigorous assessment of Chronos’s cross-domain generalization. Additional dataset details are provided in Appendix B.

**Evaluation.** We employ Pass@1 and majority voting as foundational evaluation metrics. Pass@1 measures the accuracy of a single reasoning chain, while Maj@K aggregates K parallel reasoning paths by selecting the most frequent answer, treating all trajectories as equally reliable. In addition, we include DeepConf (Fu et al., 2025) as a representative weighted voting baseline. DeepConf assigns trajectory weights based on statistical confidence estimated from token-level probability distributions, in contrast to the uniform weighting used in standard majority voting. Following the official implementation, we report the best performance achieved across mean, bottom, and tail confidence variants with varying top- $\eta$  filtering settings.

**Implementation Details.** For the AIME (2000–2023) dataset, we sample 32 complete reasoning trajectories for each question and randomly

partition them into training, validation, and test sets following an 8:1:1 ratio. In our experimental setup, we fix the input sequence length to  $L_{tail} = 2048$ , set the number of multi-scale convolutional blocks to  $N_{Blk} = 3$ , and the retention ratio  $\eta = 0.1$ . We perform a hyperparameter search for the remaining components: the projection dimension  $N_{Proj}$  is selected from {8, 16}, the number of filters  $N_{Conv}$  is searched over {4, 8, 16}, the number of distinct kernel lengths  $k = 3$ , and the set of multi-scale convolution filter lengths  $l$  is chosen from {{10, 20, 40}, {20, 40, 80}, {40, 80, 160}}.

After training Chronos, we select the hyperparameter configuration that yields the highest AUC on the test set to serve as the scorer for the final evaluation. During the evaluation phase, we first generate a candidate pool of 512 complete trajectories per question. For each experiment, we subsample 128 trajectories per question from this pool to apply the score-weighted majority voting method.

#### 4.2 Main Results (RQ1)

Table 1 summarizes the performance of Chronos in comparison with Pass@1, Majority Voting, and the confidence-weighted baseline DeepConf across three models and three benchmarks.

We highlight the key observations below:

**Superiority over Majority Voting.** As shown in Table 1, Chronos consistently surpasses the standard Majority Voting (Maj@128) across all experimental settings. While majority voting effectively reduces variance compared to Pass@1, it treats all reasoning paths as equally valid, often failing when correct answers are in the minority. Chronos addresses this by assigning quality-aware scores. For instance, on the DeepSeek-1.5B model, Chronos improves the average accuracy from 29.38% (Maj@128) to 37.09%, a substantial absolute gain of 7.71%. Similarly, on the larger DeepSeek-8B model, Chronos pushes the average accuracy from 73.54% to 78.68%. These results demonstrate that identifying and weighting high-quality trajectories is significantly more effective than simple frequency-based aggregation.

**Advantage over Uniformly Pooled Token-Level Statistics.** Chronos consistently outperforms DeepConf@128 across all nine model-dataset combinations. DeepConf aggregates token-level statistics via uniform mean pooling, treating the reasoning process as a collection of token probabilities.

<sup>1</sup>DeepSeek-1.5B: <https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B>.

<sup>2</sup>Qwen3-4B: <https://huggingface.co/Qwen/Qwen3-4B-Thinking-2507>.

<sup>3</sup>DeepSeek-8B: <https://huggingface.co/deepseek-ai/DeepSeek-R1-0528-Qwen3-8B>.

Table 1: Performance comparison. Accuracy (%) is reported. All experiments are repeated 16 times. The top two results are highlighted in **bold** and underlined.

Model	Dataset	Pass@1	Maj@128	DeepConf@128	Chronos@128
DeepSeek-1.5B	AIME25	24.58	37.50	<u>40.21</u>	<b>44.17</b>
	HMMT25	12.08	16.67	<u>25.21</u>	<b>27.92</b>
	GPQA-D	21.67	33.96	<u>36.88</u>	<b>39.17</b>
	<i>Average</i>	19.44	29.38	<u>34.10</u>	<b>37.09</b>
Qwen3-4B	AIME25	78.67	<b>86.67</b>	85.83	<b>86.67</b>
	HMMT25	55.42	60.62	<u>67.08</u>	<b>74.38</b>
	GPQA-D	59.79	<u>70.83</u>	68.96	<b>71.25</b>
	<i>Average</i>	63.96	72.71	<u>73.96</u>	<b>77.43</b>
DeepSeek-8B	AIME25	75.21	83.96	<u>86.46</u>	<b>88.75</b>
	HMMT25	60.42	68.75	<u>73.13</u>	<b>76.04</b>
	GPQA-D	59.17	67.92	<u>70.21</u>	<b>71.25</b>
	<i>Average</i>	64.93	73.54	<u>76.60</u>	<b>78.68</b>

In contrast, Chronos models each reasoning trajectory as a temporal process, and evaluates correctness by exploiting internal temporal dynamics. The consistent performance gap — for example, an improvement from 34.10% (DeepConf) to 37.09% (Chronos) on DeepSeek-1.5B — supports our hypothesis: the temporal evolution of logical consistency provides crucial discriminative signals for distinguishing correct reasoning from hallucinations, which uniform pooling fails to capture.

#### Remarkable Gains on Challenging Benchmarks.

The performance gains are particularly pronounced on complex reasoning tasks. On the HMMT25 benchmark using Qwen3-4B, Chronos achieves a striking accuracy of 74.38%, compared to 60.62% for Maj@128 and 67.08% for DeepConf. This 13.76% absolute improvement over majority voting highlights Chronos’s ability to identify correct solutions even when they are statistically underrepresented in the sample pool. Furthermore, Chronos demonstrates strong generalization on the out-of-domain GPQA-D benchmark, consistently outperforming baselines across all model scales, proving its robustness beyond mathematical domains.

### 4.3 Scaling Experiments (RQ2)

We analyze the computational efficiency of Chronos and its stability under increasing test-time compute budgets. Our analysis focuses on two key aspects: (1) the additional computational overhead introduced by the Chronos scorer and (2) performance trends as a function of the number of sampled trajectories. As illustrated in Figure 3, we examine the trade-off between computational cost

and accuracy, revealing two key observations regarding the efficiency and scalability of Chronos:

#### Strong Adaptability to Test-Time Scaling.

Chronos demonstrates exceptional adaptability to test-time scaling strategies. As the number of sampled trajectories increases, the performance of Chronos improves consistently. Notably, it does not exhibit significant diminishing returns (Chen et al., 2024) within the evaluated compute budget (ranging from 1 to 128 trajectories per question), indicating substantial potential for further performance gains as the search space expands. This robust scaling behavior suggests that Chronos effectively leverages additional compute to distinguish high-quality reasoning paths and assign higher weights to correct answers.

#### Negligible Computational Overhead.

Importantly, Chronos introduces virtually no additional computational overhead relative to the generative inference process. Chronos processes a single-channel temporal signal, and it focuses on a fixed window of the critical tail segment (*i.e.*, the final  $L_{tail}$  tokens) rather than the full sequence. On average, a forward pass of Chronos for a batch of 30 queries requires 3.9 BFLOPs. In contrast, the inference for the DeepSeek-1.5B consumes roughly 2,000 TFLOPs for the same batch. Consequently, Chronos incurs only a 0.0005% increase in total inference FLOPs, ensuring that observed scaling trends are driven almost entirely by trajectory sampling rather than scorer overhead.

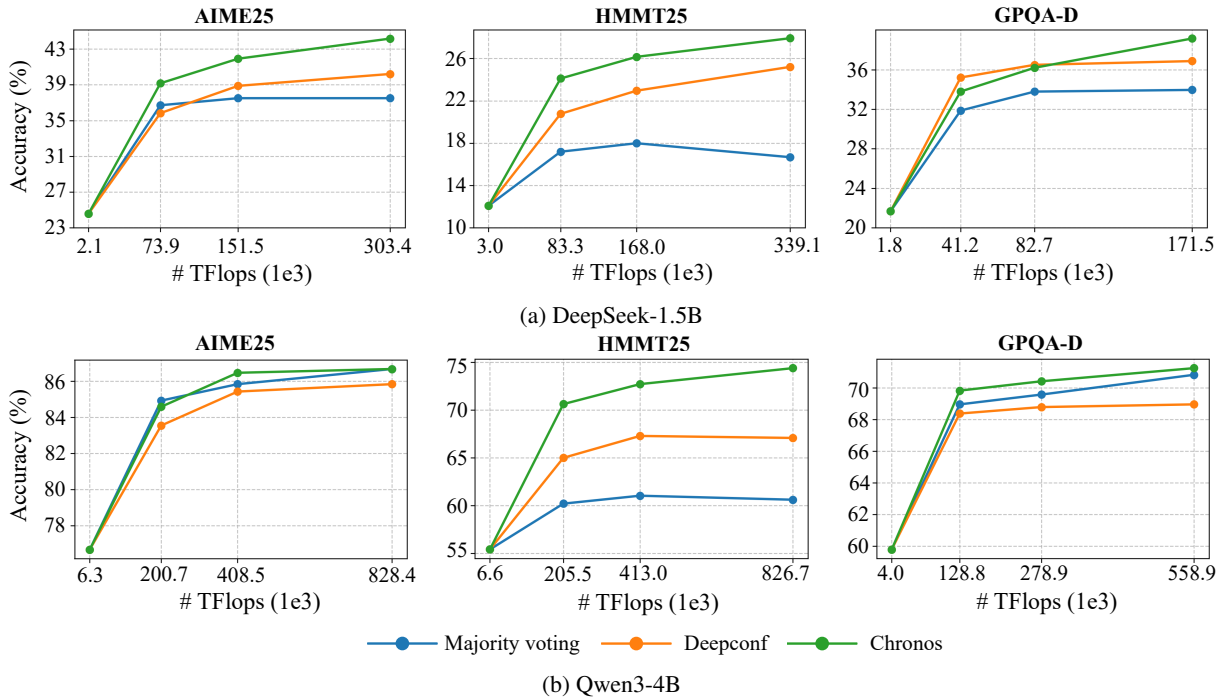


Figure 3: TTS performance on AIME25, HMMT25, and GPQA-D benchmarks. The x-axis represents the inference compute budget (# TFlops), corresponding to 1, 32, 64, 128 sampled trajectories per question, respectively.

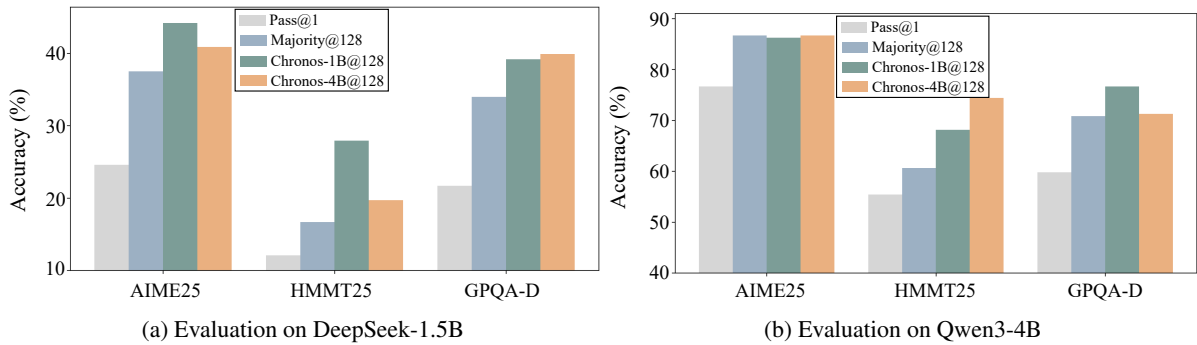


Figure 4: Cross-model generalization tests of Chronos.

#### 4.4 Cross-model Generalization Test (RQ3)

In this section, we examine whether Chronos captures model-agnostic temporal patterns of reasoning validity or primarily overfits to the distributional characteristics of a specific sampling model. To evaluate cross-model generalization, we conduct a cross-evaluation using two independently trained scorers: **Chronos-1B**, trained exclusively on trajectories generated by DeepSeek-1.5B, and **Chronos-4B**, trained on trajectories from Qwen3-4B. Each scorer is then applied to test trajectories produced by the other model.

As shown in Figure 4, Chronos demonstrates strong cross-model generalization capabilities. For instance, when applying Chronos-4B to score trajectories generated by the DeepSeek-1.5B, the performance remains highly competitive, significantly outperforming the Majority@128. Admittedly,

since distinct LLMs possess unique intrinsic reasoning logic and token-level probability distributions, a slight performance degradation is observed compared to the native in-domain scorer due to distributional shift. However, the results indicate that the temporal signal of a valid reasoning chain is largely robust across models, and Chronos successfully captures the underlying invariants. It effectively learns these model-agnostic patterns, allowing a well-trained scorer to be plugged into different LLMs without retraining, highlighting its potential as a versatile module for TTS.

#### 4.5 Further Analysis

We further conduct hyper-parameter analyses and visualize score distributions to better understand the behavior of Chronos. Due to space constraints, additional case studies are deferred to Appendix C.

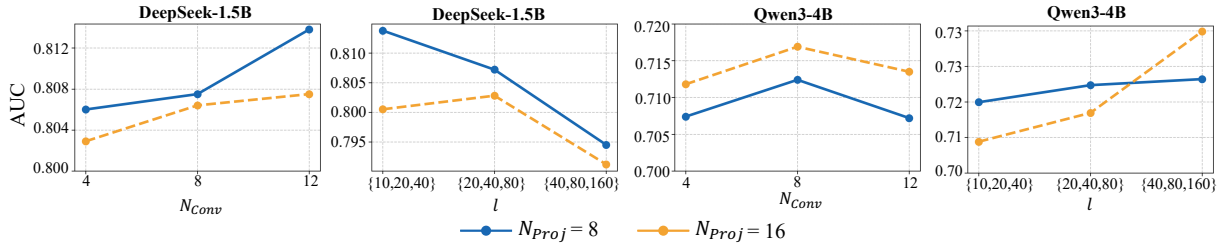


Figure 5: Hyper-parameter analysis. The plots display the AUC scores of Chronos trained and evaluated on reasoning trajectories sampled from DeepSeek-1.5B (left two panels) and Qwen3-4B (right two panels).

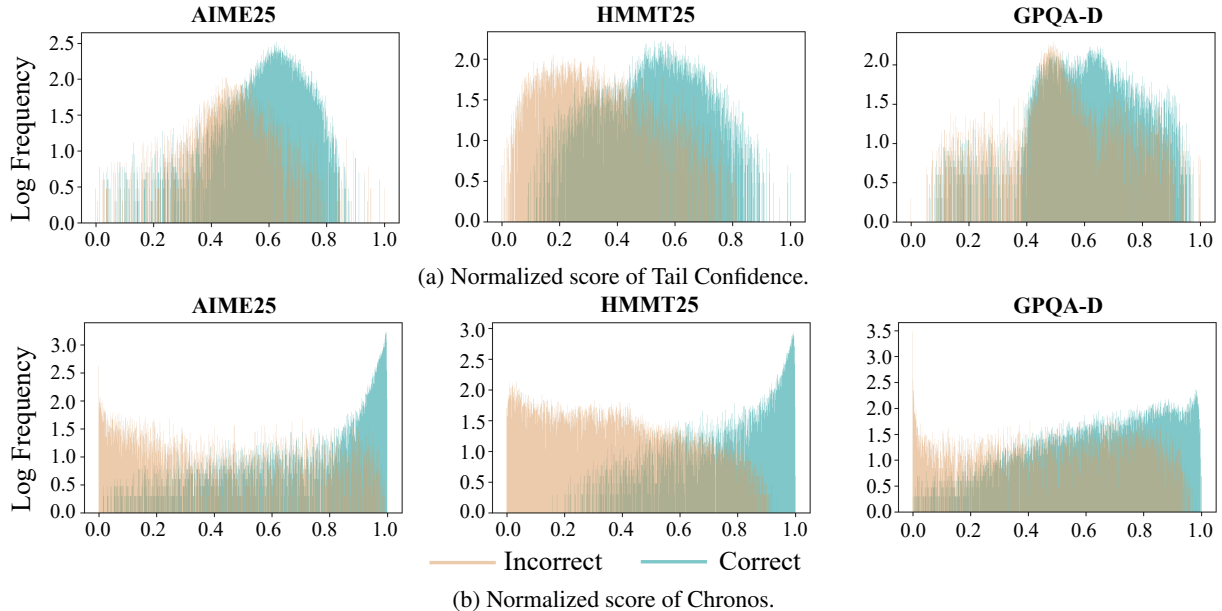


Figure 6: Comparison of normalized score distributions between Tail Confidence and Chronos on the AIME25, HMMT25, and GPQA-D benchmarks. The y-axis represents the log-scaled frequency.

**Hyper-Parameter Experiments.** We examine Chronos’s sensitivity to projection dimension  $N_{Proj}$ , the number of convolutional filters  $N_{Conv}$ , and kernel lengths  $l$ , as shown in Figure 5. On DeepSeek-1.5B trajectories, lower projection dimensions ( $N_{Proj} = 8$ ) combined with a larger number of filters ( $N_{Conv} = 12$ ) yield the best performance, suggesting that filter diversity is more critical than embedding dimensionality for smaller models. In contrast, Qwen3-4B achieves optimal performance with higher projection dimensions ( $N_{Proj} = 16$ ), indicating a greater need for expressive feature representations. With respect to temporal receptive fields, performance on DeepSeek-1.5B peaks with shorter kernel sizes ( $\{10, 20, 40\}$ ), highlighting the importance of local consistency, whereas Qwen3-4B benefits from longer kernels ( $\{40, 80, 160\}$ ), reflecting the necessity of modeling long-range dependencies in more capable models with extended reasoning chains.

**Visualization of Score Distribution.** Figure 6 visualizes the normalized score distributions for correct and incorrect reasoning trajectories. Tail

Confidence — based on static aggregation of token-level probabilities — exhibits substantial overlap between correct and incorrect trajectories, indicating high ambiguity and frequent overconfidence in erroneous reasoning. In contrast, Chronos produces a markedly more discriminative score distribution by modeling the temporal evolution of the reasoning process. This temporal modeling induces clear separation between correct and incorrect trajectories, significantly reducing ambiguity and enabling the score-weighted majority voting mechanism (Equation 10) to filter low-quality traces with higher precision. These results further confirm that Chronos captures temporal reasoning patterns that are essential for reliable trajectory quality estimation and are overlooked by rule-based metrics.

## 5 Conclusion

We introduce **Chronos**, a lightweight chronological reasoning scorer that formalizes test-time scaling by modeling inference trajectories as temporal sequences. By capturing the dynamic evolution of internal signals — specifically token-level probabil-

ities — throughout the inference process, Chronos effectively distinguishes correct reasoning from hallucinations. Our empirical evaluations demonstrate that Chronos significantly outperforms heuristic baselines across a diverse array of benchmarks and model architectures. Furthermore, our case studies confirm that chronological modeling is crucial for high-fidelity reasoning aggregation, delivering substantial performance gains with negligible computational overhead.

## 6 Limitations

While Chronos demonstrates significant efficacy in test time scaling, it necessitates white-box access to the model’s internal token-level probability distributions to construct temporal signals. Consequently, our method is currently restricted to open-weight models and precludes direct deployment on closed-source "black-box" systems (*e.g.*, proprietary API-based models) that do not expose output log-probabilities.

Furthermore, the effectiveness of Chronos is contingent upon the fundamental capacity and calibration of the underlying LLM. In scenarios where a model exhibits mode collapse or generates incoherent probability distributions devoid of logical patterns, the discriminative capability of the scorer is likely to be compromised.

Additionally, a limitation exists regarding domain specificity, as Chronos was trained exclusively on mathematical problems in this paper. Although the model demonstrates generalization to scientific benchmarks like GPQA, it remains uncertain whether the temporal signal patterns characteristic of mathematical reasoning can be effectively transferred to less structured domains, such as creative writing, the humanities, or open-ended conversational tasks.

## Ethical considerations

We caution that Chronos assesses reasoning quality based on internal model signals (*i.e.*, the temporal dynamics of the generation process), rather than factual verification. Thus, it may still validate hallucinations if they exhibit consistent internal signal patterns despite being factually incorrect. Additionally, our method relies on white-box access to these internal signals, limiting its deployment to open-weight models.

## Acknowledgement

This research is supported by the National Natural Science Foundation of China (U25A20445). This research was also supported by the advanced computing resources provided by the Supercomputing Center of the USTC.

## References

- Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. [An empirical evaluation of generic convolutional and recurrent networks for sequence modeling](#). *CoRR*, abs/1803.01271.
- Mislav Balunovic, Jasper Dekoninck, Ivo Petrov, Nikola Jovanovic, and Martin T. Vechev. 2025. Matharena: Evaluating llms on uncontaminated math competitions. *CoRR*, abs/2505.23281.
- Donald J. Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *Knowledge Discovery in Databases: Papers from the 1994 AAI Workshop, Seattle, Washington, USA, July 1994. Technical Report WS-94-03*, pages 359–370. AAAI Press.
- Bradley C. A. Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *CoRR*, abs/2407.21787.
- Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. 2024. Are more LLM calls all you need? towards scaling laws of compound inference systems. *CoRR*, abs/2403.02419.
- Zui Chen, Tianqiao Liu, Tongqing, Mi Tian, Weiqi Luo, and Zitao Liu. 2025. Advancing mathematical reasoning in language models: The impact of problem-solving data, data synthesis methods, and training stages. In *ICLR*.
- DeepSeek-AI. 2024. [Deepseek-v3 technical report](#). *CoRR*, abs/2412.19437.
- Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. In *ACL (Findings)*, pages 9367–9385.
- Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F. Schmidt, Jonathan Weber, Geoffrey I. Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. 2020. Inceptiontime: Finding alexnet for time series classification. *Data Min. Knowl. Discov.*, 34(6):1936–1962.

- Valentin Flunkert, David Salinas, and Jan Gasthaus. 2017. [Deepar: Probabilistic forecasting with autoregressive recurrent networks](#). *CoRR*, abs/1704.04110.
- Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *CoRR*, abs/2508.15260.
- Gemini-Team. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *CoRR*, abs/2507.06261.
- Siyuan Huang, Zhiyuan Ma, Jintao Du, Changhua Meng, Weiqiang Wang, and Zhouhan Lin. 2024. [Mirror-consistency: Harnessing inconsistency in majority voting](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 2408–2420. Association for Computational Linguistics.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helvar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, and 80 others. 2024. [Openai ol system card](#). *CoRR*, abs/2412.16720.
- Ehsan Kamaloo, Nouha Dziri, Charles L. A. Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *ACL (1)*, pages 5591–5606.
- Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. Scalable best-of-n selection for large language models via self-certainty. *CoRR*, abs/2502.18581.
- Tim Knappe, Ryan Li, Ayush Chauhan, Kaylee Chhua, Kevin Zhu, and Sean O’Brien. 2024. [Enhancing language model reasoning via weighted reasoning in self-consistency](#). *CoRR*, abs/2410.07839.
- Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. [Modeling long- and short-term temporal patterns with deep neural networks](#). In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pages 95–104. ACM.
- Chengpeng Li, Mingfeng Xue, Zhenru Zhang, Jiayi Yang, Beichen Zhang, Xiang Wang, Bowen Yu, Binyuan Hui, Junyang Lin, and Dayiheng Liu. 2025a. [START: self-taught reasoner with tools](#). *CoRR*, abs/2503.04625.
- Jierui Li, Hung Le, Yingbo Zhou, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. 2025b. [Codetree: Agent-guided tree search for code generation with large language models](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 3711–3726.
- Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyu Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. 2019. [Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5244–5254.
- Jessica Lin, Eamonn J. Keogh, Stefano Lonardi, and Bill Yuan-chi Chiu. 2003. [A symbolic representation of time series, with implications for streaming algorithms](#). In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, DMKD 2003, San Diego, California, USA, June 13, 2003*, pages 2–11. ACM.
- Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qixia Lai, Lingna Ma, and Qiang Xu. 2022. [Scinet: Time series modeling and forecasting with sample convolution and interaction](#). 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*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, 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*.
- OpenAI. 2025. [Introducing GPT-5](#). Accessed: 2025-08-09.
- Boris N. Oreshkin, Dmitri Carпов, Nicolas Chapados, and Yoshua Bengio. 2020. [N-BEATS: neural basis expansion analysis for interpretable time series forecasting](#). In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. [GPQA: A graduate-level google-proof q&a benchmark](#). *CoRR*, abs/2311.12022.
- Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. [Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. [Going deeper with convolutions](#). In *CVPR*, pages 1–9.

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *ICLR*.
- 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*.
- Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2025. An empirical analysis of compute-optimal inference for problem-solving with language models. In *ICLR*.
- Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P. Lillicrap, Kenji Kawaguchi, and Michael Shieh. 2024. [Monte carlo tree search boosts reasoning via iterative preference learning](#). *CoRR*, abs/2405.00451.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 40 others. 2025. [Qwen3 technical report](#). *CoRR*, abs/2505.09388.
- 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*.
- Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. [Are transformers effective for time series forecasting?](#) In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 11121–11128. AAAI Press.
- Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. [Informer: Beyond efficient transformer for long sequence time-series forecasting](#). In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The*

*Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 11106–11115. AAAI Press.

## A Experimental Settings

**Environment.** We sample a candidate pool of 512 reasoning trajectories for each question using 4\*A100 80G GPUs under the following setup:

- **vLLM:** 0.10.2;
- **Python:** 3.12.11;
- **CUDA:** 12.8.

**Sampling Parameters.** We list below the per-model decoding hyperparameters used across all experiments. For each model, we fix temperature, top- $p$ , top- $k$ , and the maximum generation length, and we use each model’s native tokenizer.

Table 2: Sampling parameters used in our experiments

Model	Temperature	Top- $p$	Top- $k$	Max seq len
DeepSeek-1.5B	0.6	0.95	20	128k
Qwen3-4B	0.6	0.95	20	128k
DeepSeek-8B	0.6	0.95	20	128k

**Prompt Templates.** For math problems (*e.g.*, AIME and HMMT), we append the following instruction to every problem prompt:

Please reason step by step, and put your final answer within `\boxed{}`.

For multiple-choice questions (*e.g.*, GPQA-D), we add the following instruction to every problem prompt:

Please reason step by step, and put your final answer within `\boxed{}`, such as `\boxed{A}`.

## B Datasets

To mitigate the risk of data leakage, we train Chronos exclusively on the AIME archive spanning 2000–2023, ensuring the training distribution maintains a difficulty level comparable to the evaluation data. For evaluation, we employ two mathematics competition datasets sourced from Math-Arena (Balunovic et al., 2025). **AIME25** constitutes the latest iteration of the American Invitational Mathematics Examination, featuring challenging high-school olympiad problems with single integer answers. **HMMT25** (Feb), from the

Harvard-MIT Mathematics Tournament, encompasses a broader spectrum of mathematical topics and frequently necessitates creative, multi-step reasoning. Each of these datasets comprises 30 examples. Furthermore, we sample 30 examples from **GPQA-Diamond** (Rein et al., 2023), which consist of graduate-level STEM reasoning tasks formatted as multiple-choice questions. Collectively, these benchmarks are widely adopted standards for evaluating frontier reasoning LLMs (*e.g.*, GPT-5 (OpenAI, 2025) and Qwen3 (Yang et al., 2025)).

## Data Splitting and Hyperparameter Selection.

The data split is performed strictly within the historical AIME 2000–2023 dataset. Because our objective is to train the model to evaluate reasoning quality based purely on the temporal dynamics of the trajectory, we perform this split at the trajectory level, remaining completely agnostic to specific question content. Most importantly, this historical dataset is entirely distinct from our final evaluation benchmarks (AIME25, HMMT25, GPQA-D) at the question level. Therefore, all data used to train and validate Chronos is strictly isolated from the final evaluation benchmarks, ensuring absolutely no data leakage. Furthermore, the hyperparameter search referenced in our main text was finalized exclusively on the held-out test subset of the AIME 2000–2023 dataset, which has no overlap with the final evaluation benchmarks. Therefore, there is no test-time tuning or information leakage affecting our reported benchmark performance.

## C Case Study

To provide an intuitive understanding of the discriminative power of Chronos, we first conduct a visualization analysis of the score distributions. As illustrated in Figure 7, we selected five distinct questions from the HMMT25 benchmark and visualized the normalized Chronos scores for trajectories generated by Qwen3-4B. For each question, we sampled 128 trajectories from a larger candidate pool of 512, repeating the experiment 16 times to ensure statistical robustness. The visualization reveals a clear polarization: correct trajectories are clustered towards the high-score region, while incorrect trajectories are suppressed into the low-score region. This distinct separation demonstrates Chronos’s ability to provide a high-fidelity signal for the subsequent weighted majority voting stage.

Furthermore, to investigate the specific temporal features captured by our model, we present a de-

Table 3: Extended baseline comparisons based on DeepSeek-R1-Distill-Qwen-1.5B.

Benchmark	Pass@1	Maj@128	SC@128	MC	DeepConf@128	Chronos@128
AIME 25	24.58	37.50	37.08	28.75	40.21	<b>44.17</b>
HMMT 25	12.08	16.67	16.67	14.17	25.21	<b>27.92</b>
GPQA-D	21.67	33.96	34.38	25.83	36.88	<b>39.17</b>
Avg	19.44	29.38	29.38	22.92	34.10	<b>37.09</b>

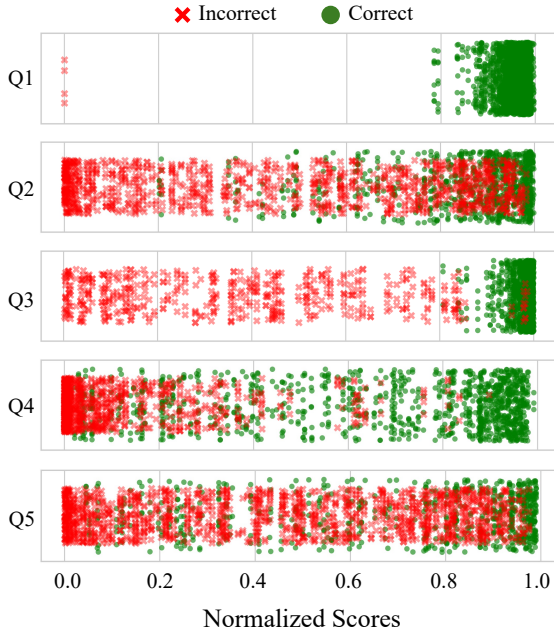


Figure 7: Question-level score distribution of Chronos based on Qwen3-4B. Q1-Q5 denotes five questions from HMMT25, respectively.

tailed comparison of two reasoning trajectories for the algebra question from HMMT25 shown above.

- **Trajectory 1 (Correct, Chronos Score: 0.8973):** It maintains a rigorous logical flow throughout the generation. In the tail phase, the model performs precise symbolic manipulations, deriving the exact values for variables  $u$  and  $t$ , and concludes with a definitive calculation of the sum.
- **Trajectory 4 (Incorrect, Chronos Score: 0.0116):** Conversely, while this trajectory starts with an identical prefix, it diverges in the latter stages. Instead of deriving the solution mathematically, the model falls into a hallucination pattern characterized by qualitative textual analysis. It discusses "rounding errors" and "numerical approximation", eventually guessing a "simplest exact expression" rather than deriving it.

It is worth noting that the text density and lack of confident symbolic derivation in the incorrect

trajectory likely manifest as distinct patterns in the token-level probability distribution. We think Chronos successfully captures these fine-grained temporal dynamics. While the correct trajectory exhibits the stable, logical signal associated with valid formulaic reasoning, the incorrect trajectory's reliance on vague verbal justifications and numerical guesses is identified as a low-quality signal. This case indicates how Chronos leverages chronological features to distinguish between rigorous reasoning and erroneous trajectories.

## D Extended Baseline Comparisons

To further strengthen our empirical evaluation, we compare Chronos against two additional aggregation baselines: weighted Self-Consistency (SC) (Wang et al., 2023) and Mirror-Consistency (MC) (Huang et al., 2024). Regarding Self-Consistency, we adopt the weighted variant that leverages token-level probability scores. Regarding Mirror-Consistency, this method relies on a serial self-refinement process rather than parallel multi-trajectory sampling. Given the 128k context length limits of reasoning models and the exceptionally long Chain-of-Thought required for complex math reasoning, serial refinement methods are heavily restricted by context windows and are not strictly comparable to parallel Test-Time Scaling approaches.

The experimental results based on DeepSeek-R1-Distill-Qwen-1.5B are presented in Table 3.

As shown, Chronos consistently outperforms all baselines across all three benchmarks. Notably, Mirror-Consistency performs even worse than standard majority voting, as the serial refinement paradigm is fundamentally constrained by context window limitations when applied to long-form mathematical reasoning.

## E AUC Analysis

We note that Chronos functions primarily as a continuous, rank-based scorer rather than a hard binary

**Question:** There exists a unique triple  $(a, b, c)$  of positive real numbers that satisfies the equations  $a^2 + 1 = 3(b^2 + 1) = 4(c^2 + 1)$  and  $a + b + c = 1$ . Compute  $a + b + c$ .

**Answer:**  $\frac{9\sqrt{23}}{23}$

**Trajectory 1 Chronos score: 0.8973**

"This is a complex or challenging question, and it is difficult to provide a direct and correct answer. I need to think about it. Well, so we have this problem where there's a unique triple  $(a, b, c)$  of positive real numbers satisfying two equations: first,  $2(a^2 + 1) = 3(b^2 + 1) = 4(c^2 + 1)$ , and second,  $ab + bc + ca = 1$ . We need to find  $a + b + c$ . Hmm, let's start by handling the first equation because it relates  $a, b, c$  through equalities, so maybe we can express each variable in terms of a common variable to simplify things ... **Omit 7,142 tokens** ... Only the positive root is valid:  $u = \frac{18}{46} = \frac{9}{23}$ . Then:  $3 - u^4 = \frac{3 - \frac{9}{23}}{4} = \frac{60}{92} = \frac{15}{23}$ . Now compute the values of  $a, b, c$ , and  $c^2 - b^2 = u = \frac{9}{23} \Rightarrow b = \frac{3\sqrt{23}}{15} - ab = t = \frac{15}{23} \Rightarrow a = \frac{t}{b} = \frac{15/23}{3\sqrt{23}/15} = \frac{5\sqrt{23}}{15}$ . From  $a^2 = 2c^2 + 1 \Rightarrow c^2 = \frac{a^2 - 1}{2} = \frac{25/23 - 1}{2} = \frac{2}{46} = \frac{1}{23} \Rightarrow c = \frac{1}{\sqrt{23}}$ . Step 4: Compute  $a + b + c = \frac{5\sqrt{23}}{15} + \frac{3\sqrt{23}}{15} + \frac{1}{\sqrt{23}} = \frac{9\sqrt{23}}{23}$ . This is the **exact** value of  $a + b + c$ , and it has been verified to satisfy all the original conditions. Final Answer:  $\frac{9\sqrt{23}}{23}$  ✓

**Trajectory 4 Chronos score: 0.0116**

This is a complex or challenging question, and it is difficult to provide a direct and correct answer. I need to think about it. Well, so we have this problem where there's a unique triple  $(a, b, c)$  of positive real numbers satisfying two equations: first,  $2(a^2 + 1) = 3(b^2 + 1) = 4(c^2 + 1)$ , and second,  $ab + bc + ca = 1$ . We need to find  $a + b + c$ . Hmm, let's start by dealing with the first equation because it relates all three variables through a common value. ... **Omit 15,468 tokens** ... The discrepancy is due to rounding errors in the numerical approximation. In fact, if we refine the value of  $S$  slightly higher (e.g.,  $S \approx 1.878$ ), the value of  $ab + bc + ca$  becomes **exactly** 1. This suggests that the **exact** value of  $a + b + c$  is:  $\sqrt{\frac{7}{2}}$ . This is the **simplest exact expression** that satisfies the conditions and matches the numerical results closely. Final Answer:  $\sqrt{\frac{7}{2}}$  ✗

Table 4: AUC comparison on the final evaluation benchmarks based on DeepSeek-R1-Distill-Qwen-1.5B. Higher AUC indicates better discrimination between correct and incorrect trajectories.

Benchmark	SC@128	DeepConf@128	Chronos@128
AIME 25	0.6765	0.7102	<b>0.8045</b>
HMMT 25	0.6462	0.7108	<b>0.7318</b>
GPQA-D	0.5961	0.6201	<b>0.7009</b>

classifier. Consequently, calculating strict classification metrics such as True Positives (TP) or False Positives (FP) is not directly applicable, as these metrics fundamentally rely on defining a rigid classification threshold. In practice, our method operates effectively without requiring such a threshold, and the baseline methods (e.g., Self-Consistency (Wang et al., 2023), DeepThink (Fu et al., 2025)) operate under the exact same rank-based paradigm.

As a more principled alternative, we report the Area Under the ROC Curve (AUC), which offers a threshold-independent and comprehensive reflection of scoring quality. AUC measures the probability that a correct trajectory receives a higher score than an incorrect one for the same problem, making it ideally suited for evaluating rank-based scorers. Table 4 presents the AUC results based on DeepSeek-R1-Distill-Qwen-1.5B. Note that we only include methods capable of assigning a continuous confidence score to individual

trajectories. Methods such as Majority Voting and Mirror-Consistency are excluded from this comparison because they operate at the population level or rely on discrete voting mechanisms, and therefore cannot produce per-trajectory scores necessary for AUC computation.

Chronos consistently achieves the highest AUC across all benchmarks, substantially outperforming both Self-Consistency and DeepThink-based confidence scoring. On AIME 25, Chronos achieves an AUC of 0.8045, representing a 13.3% absolute improvement over Self-Consistency and a 9.4% improvement over DeepConf. These results confirm that correct trajectories consistently receive higher Chronos scores than incorrect ones, validating the effectiveness of our temporal modeling approach for trajectory quality estimation.