Joint Consistency: A Unified Test-Time Aggregation Framework via Energy Minimization

Yunzhen Yao[*]
EPFL
yunzhen.yao@epfl.ch

Hongye Wang[*]
SUFE
ishongyewang@gmail.com

Yahong Wang
SUFE
2025214412@stu.sufe.edu.cn

Michael C. Gastpar
EPFL
michael.gastpar@epfl.ch

Bo Jiang[†]
SUFE
syebojiang@gmail.com

Lie He[†]
SUFE[‡]
helie@sufe.edu.cn

Abstract

This paper studies test-time aggregation, an approach that generates multiple reasoning traces and aggregates them into a final answer. Most existing methods rely on evaluation signals collected from candidate traces in isolation or answer frequencies, while ignoring comparative interactions among candidates. We propose *Joint Consistency* (JC), formulated as a constrained Ising-type energy minimization problem, which employs both independent evaluation signals and pairwise comparisons using LLM-as-a-judge. JC provides a unified framework for test-time aggregation that subsumes typical voting and weighted aggregation methods as special cases. Moreover, we develop an efficient approximation strategy that makes interaction modeling practical for large-scale test-time aggregation. Experiments on math and code reasoning benchmarks show that JC outperforms existing baselines across tasks, judge models, trace budgets, and trace-generation settings, with particularly pronounced gains on challenging problems.

1 Introduction

Scaling test-time computation is an effective approach for improving the performance of large language models (LLMs) on complex reasoning tasks. Given a specific query, Test-Time Scaling (TTS) methods allocate additional computation at inference time to improve performance. This is typically achieved by eliciting additional reasoning steps within a single response [47] or by aggregating multiple candidate responses into a final decision [39, 48, 3]. This additional computation can be executed in parallel [46] or sequentially [30, 34], and may incorporate evaluation signals at different stages of the reasoning process [52, 15].

For a wide range of TTS methods, the central question is *how to effectively aggregate diverse candidate reasoning traces into a single, reliable output*. We formulate this post-hoc decision problem as *Test-Time Aggregation* (TTA). Existing approaches to TTA include Self-Consistency [46], Weighted Self-Consistency [20], Best-of-N selection [23], Self-Certainty [25], and a broad class of evaluation-based TTS methods [10], where candidate solutions are first fully generated and subsequently evaluated, compared, and/or aggregated.

Most TTA methods evaluate reasoning traces *in isolation*: the score of each candidate is *independent* of others, disregarding comparative information across traces. This requires the evaluator to produce well-calibrated absolute quality scores, which are often difficult to obtain. In contrast, comparative

[*]Equal contribution.
[†]Corresponding authors.
[‡]MoE Key Laboratory of Interdisciplinary Research of Computation and Economics, Shanghai University of Finance and Economics (SUFE).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

judgments between candidate traces can be more reliable, as they focus on *relative* quality rather than *absolute* scores, a principle well-established in preference learning and LLM alignment [9, 33].

Nevertheless, a key question remains open: how can we incorporate pairwise relations (i.e., comparative judgments) into test-time aggregation in a principled and computationally efficient manner? It requires us to design a principled aggregation rule that leverages multiple comparative judgments, and make it scalable beyond the naive $O(N^2)$ cost of collecting all-pair comparisons.

Contributions. To address the above challenges, we propose *Joint Consistency* (JC) for Test-Time Aggregation (TTA), formulated as a *constrained energy minimization problem* with an Ising-type structure. Our contributions are threefold:

- **Modeling pairwise interactions in test-time aggregation.** Our proposed constrained energy minimization model enables JC to incorporate pairwise comparative signals through an *interaction* matrix, while also allowing independent evaluation signals to enter through an *external field* vector. Under the JC framework, we instantiate the interaction matrix using LLM-as-a-judge comparisons and consider alternative constructions, including Borda-JC.
- **A unifying and scalable formulation.** JC encompasses several existing TTA methods, including Self-Consistency and Weighted Self-Consistency, as special cases, while allowing independent and comparative evaluation signals to be combined. By varying the hyperparameter μ, JC interpolates between independent-evaluation-only and interaction-only aggregation. This perspective identifies existing TTA methods as special cases within a broader family of aggregation rules, and enables systematic exploration of stronger alternatives by changing μ. We further develop an efficient approximation strategy that retains the benefits of interaction modeling while making JC computationally scalable.
- **Consistent gains across math and code reasoning benchmarks.** We evaluate JC on large-scale reasoning traces on math and code reasoning tasks, covering both heterogeneous crowdsourced settings such as MathArena and controlled homogeneous settings. Across extensive experiments, JC outperforms existing baselines across tasks, judge models, trace budgets, and trace-generation settings, with especially large gains on challenging problems and heterogeneous trace pools. Moreover, JC scales effectively with the number of candidate traces, remains robust to hyperparameter choices, and achieves the benefits of interaction modeling at costs comparable to WSC through efficient approximation.

2 Prior Work and Preliminaries

In this section, we formulate the test-time aggregation problem and review prior work most relevant to Joint Consistency. A more comprehensive discussion of related work is provided in Appendix A.

2.1 Problem Formulation of Test-Time Aggregation

Consider a collection of N reasoning traces $\{y_i\}_{i=1}^N$, where each trace $y_i = (z_i, a_i)$ consists of a *reasoning chain* z_i and an *answer* a_i. Let $\mathcal{A} = \{a^{(1)}, \ldots, a^{(K)}\}$ denote the set of K distinct candidate answers appearing in $\{a_i\}_{i=1}^N$. The goal of Test-Time Aggregation (TTA) is to infer a single *aggregated answer* $\hat{a} \in \mathcal{A}$.

We partition the set of traces according to their answers. Formally, let $\{\mathcal{I}_k\}_{k=1}^K$ be a partition of $[N] := \{1, \ldots, N\}$ defined by

$$\mathcal{I}_k := \{\, i \in [N] \mid a_i = a^{(k)} \,\}. \tag{1}$$

Let $\mathbf{x} \in \{0,1\}^N$ be an indicator vector, where $\mathbf{x}_i = 1$ indicates that the aggregated answer $\hat{a} = a_i$. Then, the feasible set of $\mathbf{x}$ is $\{\mathbf{1}_{\mathcal{I}_k} \mid k \in [K]\}$, where $\mathbf{1}_{\mathcal{I}_k}$ denotes the indicator vector of the set $\mathcal{I}_k$.

2.2 Evaluation-Free TTA

Self-Consistency (SC). Self-Consistency [46, 6, 36, 57] is a classic *evaluation-free* method that aggregates $\{y_i = (z_i, a_i)\}_{i=1}^N$ via majority vote over final answers $\{a_i\}_{i=1}^N$, treating all candidates

with equal weight. It can be equivalently formulated as

$$\min_{\mathbf{x}\in\{\mathbf{1}_{\mathcal{I}_k}\mid k\in[K]\}} \quad -\langle \mathbf{h}, \mathbf{x}\rangle, \tag{2}$$

where $\mathbf{h} = \mathbf{1}_N \in \mathbb{R}^N$ is the all-one vector. Let $\hat{\mathbf{x}} = \mathbf{1}_{\mathcal{I}_{\hat{k}}}$ be a minimizer of (2); the aggregated answer is then $\hat{a} = a^{(\hat{k})}$.

2.3 TTA with Independent Evaluation Signals

In *evaluation-based* TTA methods, evaluation is typically applied to the complete reasoning trace, using *outcome reward models* (ORMs). Evaluation signals may be *intrinsic*, derived directly from the generator model, such as sequence log-probability, token-level certainty [24, 25, 40], or verbalized confidence [42, 50]. However, such intrinsic evaluation signals are often not directly comparable across heterogeneous generators, and are usually unavailable in crowdsourced settings, limiting their applicability. Alternatively, *extrinsic* evaluation signals can be obtained from separate verifiers [10], trained reward models [22, 41, 44], or from prompting-based evaluators such as LLM-as-a-judge [55, 18, 56, 28].

Weighted Self-Consistency (WSC). Weighted Self-Consistency extends SC (2) by encoding evaluation signals in the weight vector $\mathbf{h}$, where $\mathbf{h}_i$ is the weight of trace y_i.

Self-Certainty. Self-Certainty [25] aggregates traces based on the relative ranking of their confidence scores. However, these scores are computed *independently* for each trace, without reference to other candidates. As a result, Self-Certainty does not leverage pairwise comparison signals and can be viewed as a special case of WSC, where $\mathbf{h}_i$ is given by a Borda-style rank-based score $\mathbf{h}_i = (N - r_i + 1)^q$, where r_i denotes the rank of trace y_i according to its self-certainty score, computed by averaging the token-level KL divergence from the uniform distribution to the model's predictive distribution, and $q > 0$ is a hyperparameter.

2.4 TTA with Comparative Evaluation Signals

The key distinction between *independent* and *comparative* evaluation signals lies in whether the judge model evaluates candidates *jointly*. For instance, when employing LLM-as-a-judge, independent evaluation signals are elicited by assessing each trace in a separate query, whereas comparative evaluation signals are elicited by presenting two candidate traces together within one query context.

Knockout Tournament (KT). Knockout tournament [28] operates purely on comparative judgments. It performs aggregation through iterative pairwise comparisons, where pairwise judges determine relative correctness between candidates from distinct answer groups and progressively eliminate inferior ones until a single survivor remains. By avoiding exhaustive pairwise comparisons, this stochastic elimination procedure reduces the computational complexity to $O(N \log N)$.

Global ranking from pairwise comparisons. Rather than eliminating candidates locally, standard rank-aggregation methods summarize a collection of pairwise judgments with a scalar score for each candidate. Copeland ranks candidates by their pairwise wins and losses [14], Elo updates candidate ratings sequentially according to comparison outcomes [12], and the Bradley–Terry model fits latent scores whose differences determine pairwise preference probabilities [5]; see Section 4.1.3.

3 Joint Consistency

We propose Joint Consistency (JC), a *constrained energy minimization* approach for test-time aggregation inspired by the *Ising* model:

$$\min_{\mathbf{x}\in\{\mathbf{1}_{\mathcal{I}_k}\mid k\in[K]\}} \quad H(\mathbf{x}) := -\mu\,\langle \mathbf{h}, \mathbf{x}\rangle - \mathbf{x}^\top \mathbf{J}\mathbf{x}, \tag{3}$$

where $\mathbf{h} \in \mathbb{R}^N$ encodes independent evaluation signals as described in Section 2.3, $\mathbf{J} \in \mathbb{R}^{N\times N}$ encodes comparative evaluation signals, and $\mu \geq 0$ controls the contribution of the former.

The objective has an Ising-type structure, with $\mathbf{h}$ acting as *external fields* and $\mathbf{J}$ specifying pairwise *interactions*. Unlike the standard Ising model, the feasible set restricts $\mathbf{x}$ to the indicator vector of a

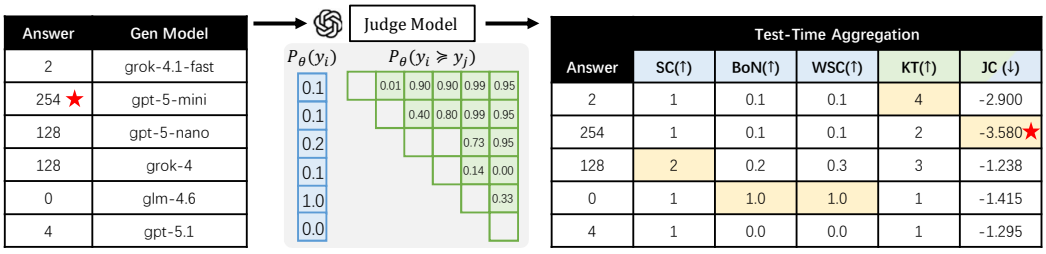

Figure 1: TTA methods on crowdsourced traces for Q28 from the HMMT 2025 (Nov) dataset, where the ground-truth answer is 254. Yellow cells denote the final aggregated answer selected by each method. Blue and green cells represent independent evaluation signals and comparative signals, respectively. The right table reports the answer-level scores used by each method: SC counts supporting traces, BoN takes the highest independent score, WSC sums independent scores, KT records the tournament round reached, and JC computes the energy in (3). Higher scores are preferred, except for JC, which minimizes energy. SC, BoN, and WSC favor incorrect majority or high-scored answers. KT eliminates the correct answer 254 in Round 2 after losing to 128 with a pairwise score of 0.4. In contrast, JC with $\mu = 0.5$ successfully identifies the correct answer by aggregating global pairwise interactions.

single answer group $\mathcal{I}_k$, as defined in (1). Figure 1 illustrates how JC leverages pairwise interactions to identify the correct answer when existing TTA methods fail.

The rest of this section is organized as follows. Section 3.1 presents our construction of $\mathbf{J}$ based on pairwise comparative judgments, along with a soft Borda alternative in Section 3.2. Section 3.3 discusses the computational complexity of Joint Consistency and introduces an efficient approximation method.

3.1 Interaction Modeling via LLM-as-a-Judge

We design the interaction matrix $\mathbf{J}$ as a positive semi-definite matrix that encodes comparative evaluation signals elicited from an LLM-as-a-judge.

Pairwise preference probabilities. Given two reasoning traces y_i and y_j, the judge model is asked to estimate the probability that y_i is better than y_j. Let π_θ denote the LLM-as-a-judge. We define

$$p_\theta(y_i \succeq y_j) := \mathbb{E}\left[P_\theta(y_i \succeq y_j)\right] \tag{4}$$

where $P_\theta(y_i \succeq y_j)$ denotes the normalized probability value output by π_θ of the event that y_i is preferred to y_j (see Appendix D.4), and the expectation is taken over the stochasticity of the judge model's responses. Concrete examples of the prompts used to query the judge model are provided in Appendix F.

Positive semi-definite construction. We construct the interaction matrix as a Gram matrix:

$$\mathbf{J} = \mathbf{C}\mathbf{C}^\top, \tag{5}$$

where $\mathbf{C} \in \mathbb{R}^{N \times N}$ has entries $C_{ij} := \sqrt{p_\theta(y_i \succeq y_j)/(n_i^2 n_j)}$ for all $i, j \in [N]$. Here, n_i denotes the number of traces sharing the same final answer as y_i; that is, if $i \in \mathcal{I}_k$, then $n_i = |\mathcal{I}_k|$. This normalization balances answer-group sizes in $\mathbf{x}^\top \mathbf{J} \mathbf{x}$: the factor n_i^2 averages over trace pairs within the selected answer group, while n_j averages over traces within each reference answer group.

Intuitively, each row of $\mathbf{C}$ can be viewed as a *comparative profile* of a trace: it represents how strongly the trace is preferred to each candidate. Consequently, $\mathbf{J}_{ij} = \langle \mathbf{C}_{i,:}, \mathbf{C}_{j,:} \rangle$ measures the similarity between the comparative profiles of y_i and y_j. Thus, JC with (5) tends to favor an answer group when its traces exhibit coherent pairwise preference patterns against the remaining candidates.

The Gram construction admits an exact decomposition into a group-balanced soft Borda score and a within-group disagreement penalty, as shown in (7). Appendix E provides a short derivation. We denote $P_\theta(y_i)$ as the independent evaluation score of trace y_i elicited by LLM-as-a-judge π_θ. The resulting Joint Consistency pipeline is summarized in Algorithm 1.

Algorithm 1 Joint Consistency for Test-Time Aggregation

1: **Input:** Question q, number of traces N, judge model π_θ, μ
2: Generate or collect N reasoning traces $\{y_i = (z_i, a_i)\}_{i=1}^N$ for question q.
3: Partition traces by their extracted answers: $\{\mathcal{I}_k\}_{k=1}^K$, where $\mathcal{I}_k = \{i : a_i = a^{(k)}\}$.
4: Estimate $\mathbf{h}$ using independent trace-level evaluation scores $P_\theta(y_i)$ from π_θ.
5: Estimate the interaction matrix $\mathbf{J}$ using pairwise comparative judgments $P_\theta(y_i \succeq y_j)$ from π_θ.
6: $\hat{\mathbf{x}} = \mathbf{1}_{\mathcal{I}_{\hat{k}}} \leftarrow$ Solve the constrained Ising objective in (3)
7: **Output:** Aggregated answer $\hat{a} = a^{(\hat{k})}$.

3.2 Alternative Constructions of the Interaction Matrix

We now present a soft Borda alternative construction of the interaction matrix $\mathbf{J}$.

Soft Borda based interaction. Writing $p_{ij} := p_\theta(y_i \succeq y_j)$, we construct a diagonal interaction matrix with

$$(\mathbf{J}_{\text{Borda}})_{ii} = \sum_{j=1}^{N} \frac{p_{ij}}{n_i n_j}. \tag{6}$$

Under $\mu = 0$, this recovers answer-level soft Borda scoring: for each answer $a^{(k)}$, the preference probabilities are averaged over trace pairs within each pair of answer groups, and these averages are then summed over the reference answer groups.

For the exact trace-level construction and $\mathbf{x} = \mathbf{1}_{\mathcal{I}_k}$,

$$\mathbf{x}^\top \mathbf{J}\mathbf{x} = \mathbf{x}^\top \mathbf{J}_{\text{Borda}}\mathbf{x} - \sum_{j=1}^{N} \frac{1}{n_j} \operatorname{Var}_{i\sim\text{Unif}(\mathcal{I}_k)}\left(\sqrt{p_{ij}}\right). \tag{7}$$

Thus, Gram-JC subtracts a penalty for disagreement among traces supporting the same answer about a fixed reference trace. The penalty is zero when traces supporting the same answer have identical preferences against each reference trace. A derivation of (7) is provided in Appendix E. Further kernel-based constructions, including RBF-JC, are presented in Appendix D.3.

3.3 Computational Complexity and Efficient Approximation

Approximation strategy. A naive implementation of JC constructs $\mathbf{J}$ by evaluating pairwise comparisons over all trace pairs, incurring an $O(N^2)$ computational cost, which becomes prohibitive when the number of traces N is large. To avoid this, we use an efficient approximation strategy: instead of exhaustively comparing all individual traces, we estimate interactions between answer groups by sampling only a constant number of traces from each group. The complexity is thus reduced from $O(N^2)$ to $O(K^2)$, where K denotes the number of distinct candidate answers.

Top-κ approximation for scalability. For challenging problems, the number of distinct answers K can still be large; see Table 8. We can therefore further restrict pairwise evaluation to the top-κ most frequent answers, where κ acts as a *consistency budget*. This reduces the pairwise comparison cost to $O(\kappa^2)$. Empirically, Figure 2 and Table 9 in Section 4.1.1 show that a small fixed κ already achieves strong performance. Answers outside the top-κ set remain eligible for selection and can still be selected if their independent evaluation scores are sufficiently high.

Cost of LLM evaluation for one trace pair. LLM-as-a-judge evaluation of a trace pair places both traces in the input prompt while requiring only a small number of output tokens to produce the judgment. As input tokens are typically substantially cheaper than output tokens, pairwise evaluation is relatively cost-efficient. For example, on OpenRouter, output tokens are approximately $4.6\times$ more expensive when querying GPT-OSS-120B ($0.18 versus $0.039 per million).[4] In contrast, reasoning trace generation often incurs a much larger number of output tokens than input tokens.

Computational tractability for solving (3)**.** In practice, the cost for solving (3) is *negligible* compared to the cost of generating reasoning traces and querying the judge model, since the feasible

[4] `https://openrouter.ai/openai/gpt-oss-120b/pricing`

set is restricted to only K configurations. With the top-κ approximation, the complexity further reduces to $O(\kappa + N)$, where N comes from the external term.

4 Experiments

We evaluate Joint Consistency on both math and code reasoning tasks. Our experiments cover two complementary settings. In the crowdsourced setting (Section 4.1), we use solutions from MathArena [4], generated by diverse models for each problem. In the homogeneous setting (Section 4.2), traces are generated by a single model, and token-level intrinsic evaluation signals, such as log-probabilities, are available. This enables comparisons with intrinsic-evaluation-based TTA baselines. Throughout the experiments, we use LLM-as-a-judge as an extrinsic evaluator for obtaining both independent and comparative evaluation signals. Unless otherwise stated, all experiments use the Gram construction of the interaction matrix $\mathbf{J}$ defined in (5); alternative constructions are explicitly labeled. Detailed experimental settings and prompt templates are provided in Appendices B and F.

4.1 Crowdsourced Tasks

We evaluate Joint Consistency on heterogeneous traces from MathArena, where candidate solutions to the same problem are generated by diverse models[5]. Since these traces lack common intrinsic evaluation signals, such as token-level log-probabilities or process rewards, methods such as Self-Certainty, DeepConf, and PRM-based aggregation are not directly applicable.

We consider four competition-level math benchmarks, namely AIME'25 [1, 2], HMMT'25-Feb, HMMT'25-Nov, and Brumo'25. These benchmarks are challenging even for strong models; for instance, GPT-4o achieves only 14% Pass@1 accuracy on HMMT'25-Feb setting, as shown in Fig. 5.

To evaluate performance across different levels of problem difficulty and generator quality, we construct three MathArena variants:

- **MathArena-Full**: All problems and available model traces.
- **MathArena-C**: A subset of challenging problems selected based on low Pass@1 accuracy and high answer diversity, measured by the average number of traces per distinct answer. Dataset statistics are provided in Table 8.
- **MathArena-W**: Traces generated by the lowest-performing models in MathArena.

4.1.1 Scalable κ-Approximation of the Interaction Matrix J

We investigate how the consistency budget κ affects the cost and accuracy of Joint Consistency on MathArena-C. Specifically, we vary the consistency budget $\kappa \in \{2, 4, 6, 8, 10\}$ and use $\{20\%, 40\%, 60\%, 80\%\}$ of the available reasoning traces for test-time aggregation. We set $\mathbf{h} \equiv 0$ to isolate the effect of the interaction matrix $\mathbf{J}$, and use GPT-OSS-20B as the judge for both JC and WSC. For a fair cost comparison, we report the actual API usage for both independent and comparative evaluations.

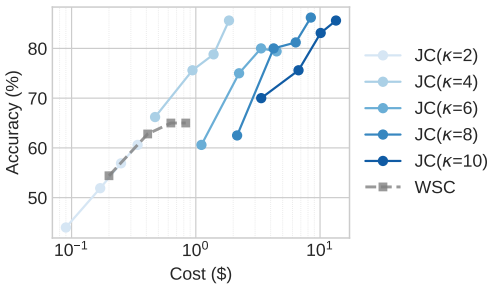

Figure 2: **Accuracy–cost trade-off under κ-approximation on MathArena-C.** Only evaluation costs are included.

The results are shown in Figure 2. We observe that **a small κ is sufficient for JC to scale beyond WSC.** JC surpasses the performance plateau of WSC and achieves an accuracy gain of over 20 percentage points, even with a small budget of $\kappa = 4$. Moreover, across different values of κ, JC continues to improve as the evaluation budget increases, suggesting that a small subset of frequent answers is sufficient to capture most of the benefit of JC. Moreover, the results show that **the interaction J can be approximated efficiently**. Detailed results are provided in Table 9 in the appendix.

[5]Traces are available at `https://huggingface.co/datasets/MathArena`. We use a revision where the four datasets are evaluated by 57, 38, 57, and 15 models, respectively. Each model provides four traces for every question.

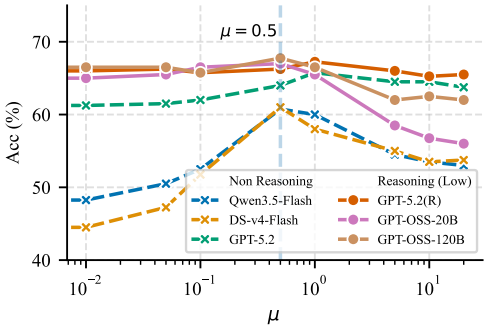

Figure 3: Sensitivity of Joint Consistency to μ on MathArena-C dataset. Across diverse judge models, JC performs best with a moderate $\mu \in [0.5, 1]$, suggesting that μ **does not require judge-model-specific tuning**.

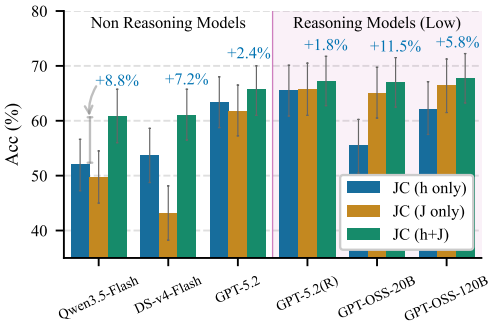

Figure 4: Accuracy of Joint Consistency across different judge models on MathArena-C dataset. Bars compare three variants of JC, namely **h** only, **J** only, and **h+J**, where the **h** only variant is WSC. The annotated values (+X%) show the accuracy gain of JC (**h+J**) over WSC, measured in percentage points.

4.1.2 Robustness to Diverse Judge Models

We evaluate Joint Consistency with diverse judge models on MathArena-C. We consider five judge model families: Qwen3.5-Flash [35], DeepSeek-V4-Flash [11], GPT-5.2 [32], GPT-OSS-20B, and GPT-OSS-120B [31]. For Qwen, DeepSeek, and the default GPT-5.2 setting, we disable reasoning. For GPT-OSS-20B and GPT-OSS-120B, we use low reasoning effort. We also include GPT-5.2 with low reasoning effort, denoted as GPT-5.2(R).

The optimal μ is stable across judge models. For each judge model, we evaluate Joint Consistency over $\mu \in \{0, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 20\}$. As shown in Figure 3, the best performance is generally achieved between 0.5 and 1, across different judges. This suggests that μ does not require careful tuning for a specific judge model.

Although GPT-5.2 with low reasoning effort achieves the strongest overall performance across most values of μ, it is $19\times$ more expensive than GPT-OSS-120B in our setting and provides only modest additional gains. This indicates that frontier closed-source models are not necessary for effective judging: open-weight GPT-OSS models offer competitive performance at substantially lower cost.

JC improves over WSC and its ablations for every judge model. As shown in Figure 4, JC *consistently* outperforms the **h**-only (WSC) and **J**-only variants across all judge models, improving over WSC by 1.8 to 11.5 percentage points. While reasoning improves pairwise evaluation and makes the **J**-only variant more competitive, combining **h** and **J** remains the most reliable choice, confirming that independent and comparative evaluation signals provide complementary information for TTA.

4.1.3 Aggregation over Complete Heterogeneous Trace Pools

We evaluate JC on MathArena-Full, which contains all available problems and model-generated traces. We compare JC and its Borda-JC variant against eight baselines: Pass@1, Best-of-N (BoN), Self-Consistency (SC), Weighted Self-Consistency (WSC), Knockout Tournament (KT), Copeland, Elo, and Bradley–Terry (BT). The three global ranking baselines use the same pairwise judgments as JC and require no additional judge calls. JC and Borda-JC are evaluated on identical trace samples and random seeds. Implementation details are provided in Appendix B.3.

JC and Borda-JC achieve comparable overall performance, with neither construction consistently outperforming the other across datasets and sampling ratios. At least one of the two achieves the best or tied-best performance in all 12 settings of Table 1. Moreover, the additional pairwise evaluation cost is significantly less than the trace-generation cost.

Table 1: **Accuracies of Joint Consistency on MathArena (Full)**. The GPT-OSS-20B model with low reasoning effort is used to score both **h** and **J**. Here, ρ denotes the fraction sampled from each dataset's trace pool.

Dataset	ρ	Pass@1	BoN	KT	SC	WSC	Copeland	Elo	BT	Borda-JC	JC
AIME 2025	10%	70.3	87.6	96.0	94.0	95.3	92.0	83.3	89.7	**96.3**	95.7
	20%	69.6	88.4	94.7	93.2	**96.3**	88.3	78.7	90.0	**96.3**	95.7
	50%	69.7	88.4	96.0	94.5	**96.7**	88.3	74.3	90.0	**96.7**	**96.7**
BRUMO 2025	10%	85.8	94.5	98.0	97.3	99.3	95.1	90.3	94.3	**100.0**	**100.0**
	20%	86.2	94.3	98.7	98.5	99.7	95.0	90.7	95.0	**100.0**	**100.0**
	50%	86.0	94.3	99.7	98.2	**100.0**	90.1	82.0	90.3	**100.0**	**100.0**
HMMT 2025 (Feb)	10%	58.4	83.9	90.7	92.7	96.2	80.7	78.7	86.3	96.3	**97.0**
	20%	58.1	82.9	94.3	94.5	96.0	80.1	81.7	84.7	96.7	**98.7**
	50%	58.3	83.4	96.0	94.0	96.7	78.3	75.3	83.7	97.0	**98.7**
HMMT 2025 (Nov)	10%	87.4	88.5	90.0	93.5	93.8	84.1	84.7	83.7	**94.3**	93.0
	20%	87.8	89.4	92.7	93.5	94.0	80.0	79.7	78.3	94.3	**95.0**
	50%	87.7	89.6	94.0	93.3	**94.3**	68.3	69.3	68.0	**94.3**	**94.3**

Performance on harder problems. To examine where aggregation gains arise, we divide each benchmark into six difficulty bins ranked by full-pool Pass@1, used only for stratification. Table 2 summarizes the hardest bin, containing the five hardest questions from each benchmark. JC improves over WSC by 2.5–4.0 percentage points across sampling ratios. Compared with Borda-JC, it is one point lower at $\rho = 10\%$ but 2.0 and 2.5 points higher at the two larger ratios. The other five bins are largely saturated, with JC and WSC differing by at most one point. These results suggest that the benefit is concentrated on harder problems in this trace pool. Full results are provided in Appendix D.2.

Table 2: **Accuracy on the hardest MathArena-Full difficulty bin.** Results average over 20 questions (five per benchmark) and the same ten seeds as Table 1. Δ_{WSC} and Δ_{Borda} are JC's gains in percentage points. Bold marks the highest accuracy in each row.

ρ	WSC	Borda-JC	JC	Δ_{WSC}	Δ_{Borda}
10%	78.0	**81.5**	80.5	+2.5	−1.0
20%	80.0	82.0	**84.0**	+4.0	+2.0
50%	81.5	82.0	**84.5**	+3.0	+2.5

4.1.4 Robustness to Lower-Tier Trace Generators

We investigate whether JC remains effective when reasoning traces are generated by lower-performing models. We rank models by their Pass@1 accuracy on AIME 2025, retain the R lowest-performing models, and evaluate $R \in \{3, 5, 10, 15, 20, 30\}$.

JC remains effective with lower-tier trace generators. As shown in Table 3, JC and Borda-JC outperform the competing baselines on average, both achieving 61.89%, compared with 56.91% for the strongest non-JC baseline, WSC. Borda-JC performs best at $R = 20$ and $R = 30$. These results suggest that JC can effectively aggregate reasoning traces even when they are generated by relatively weak models.

4.2 Comparing Intrinsic-Evaluation TTA Baselines in Homogeneous Settings

We now consider homogeneous settings where traces are generated together with token-level log-probabilities by the same model, making intrinsic-evaluation-based TTA methods applicable. We thus add two representative baselines: Self-Certainty [25] implemented with Borda voting with exponent 2, and DeepConf [15] implemented with tail confidence. We evaluate this setting on math reasoning tasks and code reasoning tasks.

For math reasoning, we conduct experiments on AIME 2025. We use Qwen3-8B, Qwen3-32B [51], and DeepSeek-R1-8B [19] as trace generators, and DeepSeek-R1-8B and GPT-OSS-20B [31] as judge models. For each problem, we generate an initial pool of 256 reasoning traces and uniformly subsample 64 traces for aggregation. Detailed hyperparameters are provided in Appendix B. As shown in Table 4, JC outperforms all other baselines across all trace generators and judge models.

Table 3: **Accuracies of TTA methods on traces from lower-tier models on AIME 2025.** R denotes the R lowest-ranked AIME 2025 generators. We use 50%-per-problem sampling. Copeland, Elo, soft-outcome Bradley–Terry (BT), Borda-JC and JC use the complete answer-pair comparison graph; KT uses pairwise tournament elimination. Entries report mean accuracies, and the final row is the average over the six values of R; the corresponding standard-deviation statistics are reported in Appendix Table 14.

R	Pass@1	BoN	KT	SC	WSC	Copeland	Elo	BT	Borda-JC	JC
3	11.2	23.7	9.3	19.6	23.5	26.7	24.3	**27.0**	**27.0**	**27.0**
5	16.0	31.8	13.6	29.5	31.8	34.3	32.0	**36.3**	**36.3**	**36.3**
10	26.9	50.0	40.9	43.0	51.6	59.3	46.3	**60.0**	58.0	59.7
15	34.3	63.1	57.9	54.7	68.7	69.2	48.7	**76.3**	74.0	76.0
20	40.9	67.3	69.4	71.5	77.7	70.8	52.7	70.7	**84.3**	82.3
30	52.1	76.5	83.6	83.3	88.2	63.7	44.7	60.7	**91.7**	90.0
Average	30.23	52.07	45.78	50.27	56.91	54.00	41.44	55.17	**61.89**	**61.89**

Table 4: **TTA accuracy on AIME 2025.** JC outperforms intrinsic-evaluation baselines across trace generators and judge models.

Judge Models	TTA Methods	Trace Generation Models: Qwen3 8B	Qwen3 32B	DS-R1 8B
-	SC	77.0	80.2	83.1
	Self-Cert.	76.6	80.0	83.1
	DeepConf	76.8	80.3	83.1
DS-R1 8B	WSC	76.8	80.7	85.5
	JC	**84.1**	**84.9**	**87.1**
GPT-OSS 20B	WSC	81.4	83.4	88.4
	JC	**86.2**	**90.9**	**93.5**

Table 5: **TTA accuracy on CruxEval-O code reasoning.** Δ denotes JC's gain over the strongest non-JC baseline.

Method	N=4	N=8	N=16	N=24
Pass@1	71.3	71.3	71.3	71.3
BoN	76.2	77.0	78.0	78.0
SC	74.7	76.0	76.3	76.4
WSC	<u>76.9</u>	<u>77.8</u>	<u>78.1</u>	<u>78.1</u>
Self-Cert.	73.5	74.7	75.6	76.0
DeepConf	71.9	74.7	75.6	75.8
JC	**77.3**	**78.7**	**79.9**	**80.5**
Δ	+0.4	+0.9	+1.8	+2.4

For code reasoning problems, we evaluate on CruxEval-O [17], a benchmark in which the model is given a Python function and an input and must predict the corresponding program output. The dataset contains 800 questions. We use Qwen3-Coder-30B to generate 32 traces for each question and resample $N \in \{4, 8, 16, 24\}$ traces for test-time scaling. To make comparison fair, we constrain the computational budget of JC to scale linearly with N. The results are shown in Table 5, from which we observe that JC outperforms all baselines for every N. Moreover, the gain Δ over the strongest baseline increases with N, indicating that JC benefits more effectively from larger test-time budgets.

The above results show that **JC remains effective even when competing baselines have access to intrinsic evaluation signals, and continues to scale favorably as the number of traces increases.**

5 Conclusion

We introduced Joint Consistency, a unified test-time aggregation framework that integrates independent evaluations with pairwise comparative signals under a constrained Ising formulation. JC highlights interaction modeling as an important component that is largely absent from existing TTA methods. Our experiments showed that JC outperforms strong baselines in both crowdsourced and homogeneous settings, while incurring only marginal additional cost via our efficient approximation strategy.

Limitations. This work has several limitations. First, our paper focuses on final-answer aggregation in reasoning tasks, and it remains unclear to what extent the observed improvements generalize to open-ended reasoning or research-level problem solving. Second, although we provide a theoretically motivated construction of the interaction matrix $\mathbf{J}$ in Section 3.1, this design is not necessarily optimal.

Exploring alternative constructions of $\mathbf{J}$ remains an important direction for future work. Finally, our method may inherit the limitations and potential risks of the underlying generation and judge models.

Acknowledgments and Disclosure of Funding

Michael Gastpar was supported by the Swiss National Science Foundation (SNSF) under Grants 236494 and 10004381. Bo Jiang was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 72394364 and 72394360. Lie He was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62606302 and by IQuest Research.

References

[1] Art of Problem Solving. 2025 AIME I. Art of Problem Solving Wiki, 2025. Accessed: 2025.

[2] Art of Problem Solving. 2025 AIME II. Art of Problem Solving Wiki, 2025. Accessed: 2025.

[3] Vidhisha Balachandran, Jingya Chen, Lingjiao Chen, Shivam Garg, Neel Joshi, Yash Lara, John Langford, Besmira Nushi, Vibhav Vineet, Yue Wu, et al. Inference-time scaling for complex tasks: Where we stand and what lies ahead. *arXiv preprint arXiv:2504.00294*, 2025.

[4] Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmark*, 2025.

[5] Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

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

[7] Jiefeng Chen, Jie Ren, Xinyun Chen, Chengrun Yang, Ruoxi Sun, Jinsung Yoon, and Sercan Ö Arık. Sets: Leveraging self-verification and self-correction for improved test-time scaling. *arXiv preprint arXiv:2501.19306*, 2025.

[8] Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language models. In *ICML Workshop on In-Context Learning*, 2024.

[9] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 2017.

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

[11] DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

[12] Arpad E. Elo. *The Rating of Chessplayers, Past and Present*. Batsford, London, 1978.

[13] Thomas Palmeira Ferraz, Kartik Mehta, Yu-Hsiang Lin, Haw-Shiuan Chang, Shereen Oraby, Sijia Liu, Vivek Subramanian, Tagyoung Chung, Mohit Bansal, and Nanyun Peng. Llm self-correction with decrim: Decompose, critique, and refine for enhanced following of instructions with multiple constraints. In *Findings of the Association for Computational Linguistics: EMNLP*, 2024.

[14] Peter C. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977.

[15] Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025.

[16] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024.

[17] Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I Wang. Cruxeval: A benchmark for code reasoning, understanding and execution. *arXiv preprint arXiv:2401.03065*, 2024.

[18] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *The Innovation*, 2024.

[19] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[20] Jiaxin Guo, Zewen Chi, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Reward reasoning models. In *Neural Information Processing Systems*, 2025.

[21] Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

[22] Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J Foster. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment. In *Forty-second International Conference on Machine Learning*, 2025.

[23] Yuu Jinnai, Tetsuro Morimura, Kaito Ariu, and Kenshi Abe. Regularized best-of-n sampling with minimum bayes risk objective for language model alignment. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2025.

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

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

[26] Tim Knappe, Ryan Luo Li, Ayush Chauhan, Kaylee Chhua, Kevin Zhu, and Sean O'Brien. Semantic self-consistency: Enhancing language model reasoning via semantic weighting. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS*, 2024.

[27] Kuang-Huei Lee, Ian Fischer, Yueh-Hua Wu, Dave Marwood, Shumeet Baluja, Dale Schuurmans, and Xinyun Chen. Evolving deeper llm thinking. *arXiv preprint arXiv:2501.09891*, 2025.

[28] Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. Pairjudge rm: Perform best-of-n sampling with knockout tournament. *arXiv preprint arXiv:2501.13007*, 2025.

[29] Jieyi Long. Large language model guided tree-of-thought. *arXiv preprint arXiv:2305.08291*, 2023.

[30] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 2023.

[31] OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025.

[32] OpenAI. Introducing gpt-5.2. `https://openai.com/index/introducing-gpt-5-2/`, December 2025. Accessed: 2026-05-05.

[33] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 2022.

[34] Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. Recursive introspection: Teaching language model agents how to self-improve. *Advances in Neural Information Processing Systems*, 2024.

[35] Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026.

[36] Rylan Schaeffer, Joshua Kazdan, John Hughes, Jordan Juravsky, Sara Price, Aengus Lynch, Erik Jones, Robert Kirk, Azalia Mirhoseini, and Sanmi Koyejo. How do large language monkeys get their power (laws)? In *Forty-second International Conference on Machine Learning*, 2025.

[37] Zhenning Shi, Yijia Zhu, Yi Xie, Junhan Shi, Guorui Xie, Haotian Zhang, Yong Jiang, Congcong Miao, and Qing Li. Reasoning under uncertainty: Efficient llm inference via unsupervised confidence dilution and convergent adaptive sampling. In *Conference on Empirical Methods in Natural Language Processing*, 2025.

[38] Harman Singh, Xiuyu Li, Kusha Sareen, Monishwaran Maheswaran, Sijun Tan, Xiaoxia Wu, Junxiong Wang, Alpay Ariyak, Qingyang Wu, Samir Khaki, et al. V1: Unifying generation and self-verification for parallel reasoners. In *The 1st Workshop on Scaling Post-training for LLMs*, 2026.

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

[40] Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in llms. *arXiv preprint arXiv:2502.06233*, 2025.

[41] Ramya Keerthy Thatikonda, Wray Buntine, and Ehsan Shareghi. Logical reasoning with outcome reward models for test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26113–26123, 2025.

[42] Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

[43] Shubham Toshniwal, Ivan Sorokin, Aleksander Ficek, Ivan Moshkov, and Igor Gitman. Genselect: A generative approach to best-of-n. In *2nd AI for Math Workshop@ ICML 2025*, 2025.

[44] Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. Reasoning aware self-consistency: Leveraging reasoning paths for efficient llm sampling. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3613–3635, 2025.

[45] Ziyu Wan, Xidong Feng, Muning Wen, Stephen Marcus Mcaleer, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like tree-search can guide large language model decoding and training. In *International Conference on Machine Learning*, 2024.

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

[47] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[48] Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. From decoding to meta-generation: Inference-time algorithms for large language models. *Transactions on Machine Learning Research*, 2024.

[49] Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023.

[50] Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *International Conference on Learning Representations*, 2024.

[51] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[52] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 2023.

[53] Jiaxin Zhang. Confidence-aware reasoning: Optimizing self-guided thinking trajectories in large reasoning models. In *Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2025.

[54] Eric Zhao, Pranjal Awasthi, and Sreenivas Gollapudi. Sample, scrutinize and scale: Effective inference-time search by scaling verification. In *Forty-second International Conference on Machine Learning*, 2025.

[55] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 2023.

[56] Yilun Zhou, Austin Xu, PeiFeng Wang, Caiming Xiong, and Shafiq Joty. Evaluating judges as evaluators: The jetts benchmark of llm-as-judges as test-time scaling evaluators. In *International Conference on Machine Learning*, 2025.

[57] Zhi Zhou, Yuhao Tan, Zenan Li, Yuan Yao, Lan-Zhe Guo, Yu-Feng Li, and Xiaoxing Ma. A theoretical study on bridging internal probability and self-consistency for llm reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

A Related Work

Test-time scaling (TTS) improves accuracy on complex reasoning tasks for LLMs by allocating additional compute for sampling and aggregating candidate solutions into a final decision [39, 48, 3]. Existing TTS methods can be categorized along three aspects. First, depending on whether evaluation is employed, a TTS method can be *evaluation-free* or *evaluation-based*. Second, evaluation can be applied to the complete reasoning trace, known as *outcome reward models* (ORMs), or to intermediate reasoning steps, corresponding to *process reward models* (PRMs). Finally, TTA can be implemented *in parallel*, where candidate solutions are generated independently and aggregated post hoc, or *sequentially*, where generation, evaluation, and refinement are interleaved.

Self-Consistency (SC) [46, 6, 36, 57] is a classic *evaluation-free* method that aggregates multiple sampled reasoning paths via majority voting over final answers, treating all candidates with equal weight. Subsequent work extends SC by incorporating more structured aggregation mechanisms. Semantic self-consistency [26] replaces string-level voting with semantic-aware clustering and aggregation while universal self-consistency [8] generalizes SC to free-form generation by prompting the LLM to select the final answer most consistent with the overall set.

Besides the most relevant comparative evaluation-based methods discussed in Section 2.4, some papers employ listwise and adaptive comparison. GenSelect [43] performs a listwise comparison by placing multiple candidate traces in a shared context and directly selecting the best candidate. V_1-Infer [38] instead uses pairwise comparisons adaptively: it first collects enough comparisons to estimate initial candidate scores and then compares similarly ranked candidates in a Swiss-tournament stage.

PRM-based methods incorporate state- or step-level evaluation directly into the generation process, actively guiding reasoning via inference-time search algorithms. In Tree of Thoughts (ToT) [52, 29], intermediate reasoning states are evaluated either using task-specific heuristics or by prompting the LLM itself to assess the path promise. Monte Carlo Tree Search (MCTS) based methods [21, 45] use step-level value estimation to prune unpromising branches and navigate the reasoning space. More recent work explores confidence-aware generation strategies, where intrinsic uncertainty estimates are used to dynamically control reasoning depth and trajectories during inference [53, 15]. PRM-based methods rely on generation-time signals and thus fall outside the scope of the post-hoc aggregation setting, especially in crowdsourced scenarios where such information is typically unavailable.

In contrast to PRM-based methods that assess intermediate reasoning steps, ORM-based methods weight or rank candidate solutions using *solution-level evaluation* signals. These signals may be *intrinsic*, derived directly from the generator model, such as sequence log-probability, token-level certainty [24, 25, 40], or verbalized confidence [42, 50]. However, such intrinsic evaluation signals are often not directly comparable across heterogeneous generators, and are usually unavailable in crowdsourced settings, limiting their applicability. Alternatively, *extrinsic* evaluation signals can be obtained from separate verifiers [10], trained reward models [22, 41, 44], prompting-based evaluators such as LLM-as-a-judge [55, 18, 56, 28]. Beyond fixed-budget parallel aggregation, some ORM-based methods adopt adaptive sampling strategies, where evaluation statistics computed over previously sampled candidates are used to dynamically determine when to stop further sampling, thus improving inference efficiency [37].

Sequential ORM-based approaches iteratively verify, critique, and refine candidate solutions using evaluation feedback, progressively steering the reasoning process toward correctness [7]. Representative methods include self-verification [30, 49, 13, 54] and self-correction [34, 16, 27].

B Hyperparameter Settings

The experiments were conducted on a server with an AMD EPYC 7402 24-Core processor, 256 GB of memory, and four NVIDIA RTX A6000 GPUs, running Ubuntu 22.04.5 LTS and Python 3.12.11.

B.1 Hyperparameters of Trace Generation Models

Table 6 summarizes the decoding hyperparameters used during generation for all models. For each model, the temperature, and maximum generation length are fixed across experiments, and we use the model's native tokenizer. A dash (—) denotes that the corresponding decoding option is

disabled. The Reasoning budget (API param) column specifies how each model's reasoning behavior is controlled during generation. For models that expose explicit API-level controls, we report the corresponding parameter setting (e.g., `reasoning_effort` for GPT-OSS). For models without a dedicated reasoning parameter, reasoning is enabled either by default or via prompt-level instructions.

Table 6: Hyperparameters and statistics of trace generation models.

Model	Task	Temperature	Max seq len	Reasoning budget (API param)
Qwen3-8B	Tab. 4	0.6	40k	True (`enable_thinking`)
Qwen3-32B	Tab. 4	0.6	40k	True (`enable_thinking`)
DeepSeek-R1-8B	Tab. 4	0.6	130k	—
Qwen3-Coder-30B	Tab. 5	0.7	256k	—

B.2 Hyperparameters of Judge Models

Table 7 summarizes the decoding hyperparameters used during scoring for all models. For each model, the temperature, and maximum generation length are fixed across experiments, and we use the model's native tokenizer. The Reasoning budget (API param) column specifies how each model's reasoning behavior is controlled during generation. For models that expose explicit API-level controls, we report the corresponding parameter setting (e.g., `reasoning_effort` for GPT-OSS). For models without a dedicated reasoning parameter, reasoning is enabled either by default or via prompt-level instructions.

Table 7: Hyperparameters and statistics of judge models.

Model	Task	Temperature	Max seq len	Reasoning budget (API param)
DeepSeek-R1-8B	Tab. 4	0.7	40k	—
GPT-OSS-20B	Tab. 1, 3, 4, 5, 9, Fig. 2, 3, 4	1.0	128k	low (`reasoning_effort`)
GPT-OSS-120B	Fig. 3, 4	0.7	128k	low (`reasoning_effort`)
GPT-5.2	Fig. 3, 4	0.7	400k	w/, w/o low (`reasoning_effort`)
Qwen3.5-Flash	Fig. 3, 4	0.7	1000k	w/o
DeepSeek-V4-Flash	Fig. 3, 4	0.7	1000k	w/o

B.3 Hyperparameters and Settings of Aggregation Methods

Tables 1 and 3 include three global ranking baselines: Copeland, Binary Elo, and Bradley–Terry (BT). In Table 1, they use the top-4 answer filter; in Table 3, they use the complete answer-pair comparison graph. In both cases, they reuse the pairwise judgments available to JC, operate on candidate answers, and require no additional judge calls.

Copeland ranking. For each pair of candidate answers, we average the available pairwise preference probabilities in each direction. A candidate receives a win if its mean preference exceeds that of its opponent, a loss if it is lower, and a tie otherwise. The Copeland score of candidate i is $C_i = W_i - L_i$, where W_i and L_i denote its numbers of wins and losses against distinct opponents. Candidates are ranked by their Copeland scores. This method is order-invariant but discards preference magnitudes.

Binary Elo. We initialize each candidate's rating to 1000 and convert pairwise preference probabilities into binary outcomes: $S_i = 1$ for a win, $S_i = 0$ for a loss, and $S_i = 0.5$ for a tie. We process one deterministically shuffled pass over all comparisons. For candidates i and j, the expected outcome is

$$E_i = \frac{1}{1 + 10^{(R_j - R_i)/400}},$$

and the ratings are updated as

$$R_i \leftarrow R_i + 400(S_i - E_i), \qquad R_j \leftarrow R_j - 400(S_i - E_i).$$

Candidates are ranked by their final ratings. Unlike Copeland, Elo depends on the order of comparisons, and its binary outcomes discard preference magnitudes.

Bradley–Terry ranking. We assign each candidate i a latent strength θ_i and model its pairwise preference probability as

$$P(i \succ j) = \sigma(\theta_i - \theta_j),$$

where σ is the logistic function. We use the same bidirectionally normalized soft preference $q_{ij} \in [0, 1]$ as JC and estimate the latent strengths by minimizing the fractional binary cross-entropy

$$-\sum_{(i,j)} \left[q_{ij} \log \sigma(\theta_i - \theta_j) + (1 - q_{ij}) \log\big(1 - \sigma(\theta_i - \theta_j)\big) \right] + \frac{\lambda}{2} \|\theta\|_2^2,$$

where $\lambda = 0.01$. We use BFGS for optimization and rank candidates by their fitted latent strengths. Unlike Copeland and Binary Elo, BT retains preference magnitudes and is independent of comparison order.

C Dataset Statistics

C.1 Pass@1 Accuracy of LLM-as-a-Judge

Figure 5 shows Pass@1 accuracies of reasoning traces from 57 models on the HMMT-2025 (Feb), measuring the effectiveness of different judge models by scoring traces.

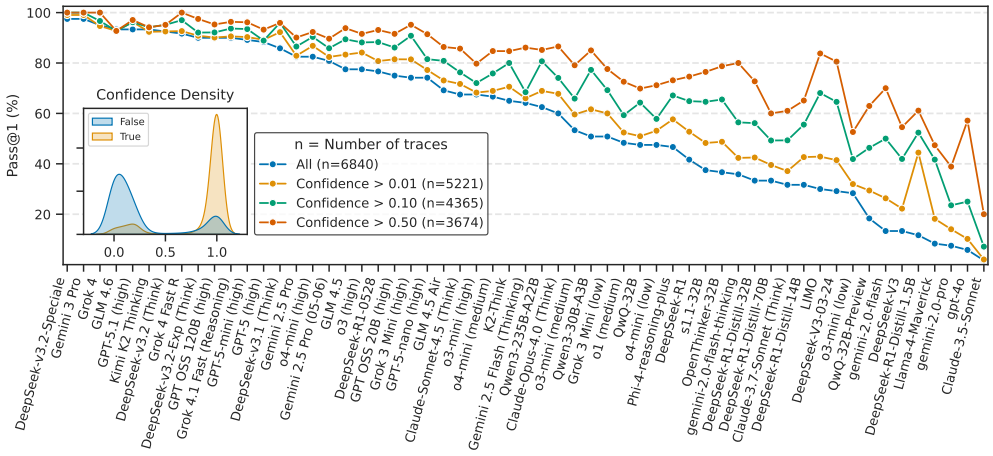

Figure 5: Pass@1 accuracies of reasoning traces from 57 models on the HMMT-2025 (Feb). Confidence scores for these traces are evaluated using a GPT-OSS-20B model with low reasoning effort. The inset plot illustrates the score density functions for correct versus incorrect reasoning traces across the full set of 6,840 traces.

C.2 Statistics of MathArena-C

Questions used in MathArena-C are detailed in Table 8. The Pass@1 accuracy is calculated across all available models in MathArena.

Table 8: Statistics for the challenging subset of questions. Higher values (↑) for #Traces/#Ans and Pass@1 typically indicate *lower* problem difficulty.

Dataset	Q Idx	#Traces/#Ans (↑)	Pass@1 (↑)
AIME	14	228/75≈3.0	13.6%
	15	228/78≈2.9	3.5%
	All	Avg≈7.0	69.8%
BRUMO	13	152/20≈7.6	46.1%
	30	152/48≈3.2	21.1%
	All	Avg≈14.7	86.1%
HMMT Feb	19	228/54≈4.2	10.5%
	20	228/103≈2.2	12.7%
	All	Avg≈5.8	60.0%
HMMT Nov	10	60/11≈5.5	6.7%
	28	60/20=3.0	11.7%
	All	Avg≈14.4	87.8%

D Additional Experimental Results

D.1 Cost and Accuracy with Interaction Matrix Simplification

Table 9 lists the data used to generate Figure 2.

Table 9: **Accuracy (%) and cost (USD) across varying consistency budgets (κ) and trace subsample ratio (ρ) on challenging crowdsourced problems.** Each cell displays the format "Accuracy(%)/$Cost". Red text denotes the highest accuracy achieved for a specific ρ. Cells highlighted with a green background indicate configurations that outperform the best overall WSC accuracy (65%). Note that while WSC is theoretically independent of κ, minor fluctuations in its cost and performance are due to the sampling stochasticity of the traces.

ρ	0.20		0.40		0.60		0.80	
κ	JC ($\mathbf{h} \equiv \mathbf{0}$)	WSC ($\mathbf{J} \equiv \mathbf{0}$)	JC ($\mathbf{h} \equiv \mathbf{0}$)	WSC ($\mathbf{J} \equiv \mathbf{0}$)	JC ($\mathbf{h} \equiv \mathbf{0}$)	WSC ($\mathbf{J} \equiv \mathbf{0}$)	JC ($\mathbf{h} \equiv \mathbf{0}$)	WSC ($\mathbf{J} \equiv \mathbf{0}$)
2	44.0%/$0.09	54.4%/$0.20	51.9%/$0.17	62.8%/$0.41	56.9%/$0.25	65.0%/$0.63	60.6%/$0.34	65.0%/$0.83
3	42.9%/$0.25	57.2%/$0.22	52.5%/$0.48	63.7%/$0.41	58.8%/$0.74	63.7%/$0.63	60.0%/$1.00	62.5%/$0.83
4	66.2%/$0.47	63.1%/$0.22	75.6%/$0.94	62.8%/$0.42	78.8%/$1.39	64.4%/$0.63	85.6%/$1.85	62.5%/$0.83
5	68.1%/$0.72	59.1%/$0.21	78.1%/$1.49	62.5%/$0.42	80.0%/$2.21	64.4%/$0.64	85.6%/$2.95	63.1%/$0.83
6	60.6%/$1.11	57.5%/$0.21	75.0%/$2.23	64.4%/$0.42	80.0%/$3.35	63.1%/$0.62	79.4%/$4.47	65.0%/$0.83
8	62.5%/$2.15	57.5%/$0.22	80.0%/$4.23	62.8%/$0.41	81.2%/$6.37	65.0%/$0.64	86.2%/$8.43	63.7%/$0.83
10	70.0%/$3.36	58.3%/$0.21	75.6%/$6.71	63.4%/$0.42	83.1%/$10.12	64.4%/$0.61	85.6%/$13.44	63.1%/$0.84
12	61.9%/$4.66	59.4%/$0.20	76.9%/$9.36	63.1%/$0.41	80.0%/$14.19	63.7%/$0.61	80.6%/$18.87	63.1%/$0.83
14	58.8%/$5.76	59.4%/$0.21	74.4%/$11.52	64.7%/$0.42	71.2%/$17.58	63.7%/$0.63	67.5%/$23.35	63.7%/$0.83
16	61.3%/$6.85	60.9%/$0.20	71.2%/$13.73	63.7%/$0.41	65.0%/$20.96	64.4%/$0.62	64.4%/$27.79	62.8%/$0.84
18	63.1%/$8.23	61.6%/$0.21	68.1%/$16.43	63.7%/$0.43	70.6%/$25.35	63.7%/$0.63	65.6%/$33.53	64.4%/$0.83
20	53.8%/$9.50	59.1%/$0.21	70.0%/$18.93	60.6%/$0.42	66.9%/$29.29	63.7%/$0.63	63.1%/$38.75	62.5%/$0.83

We also examine the effect of κ when correct answers are infrequent, using traces from lower-tier generators. Table 10 lists the number of questions out of 30 questions such that the number of traces of the correct answer is within top-κ frequent traces generated by the R low-tier models in column 2-4. The last column shows the number of questions for which the correct answer appears among the traces generated by the R models.

Table 10: Number of questions (out of 30) for which the correct answer appears among the top-κ most frequent answers in traces generated by the R low-tier models. "All" considers all generated answers.

R	Top-1	Top-4	Top-8	All
3	7	8	10	11
5	9	12	12	13
10	13	17	22	22
15	17	23	25	27
20	24	25	25	28
30	25	27	29	29

Table 11 compares WSC with JC on traces from the R lowest-ranked generators.

Table 11: Accuracy comparisons across values of R and consistency budgets κ.

R	WSC	$\kappa = 4$	$\kappa = 8$	$\kappa = K$
3	23.5	23.5	25.7	27.0
5	31.8	36.0	36.3	36.3
10	51.6	50.7	61.3	59.7
15	68.7	73.0	79.0	76.0
20	77.7	82.7	83.0	82.3
30	88.2	89.7	94.0	90.0

D.2 Performance across Problem Difficulty

We examine whether the differences in Table 1 are concentrated at a particular difficulty level. Within each benchmark, we rank its 30 questions by Pass@1 over the full trace pool and divide them into six equal bins of five questions. Each row of Table 12 therefore pools 20 questions, with every benchmark contributing equally, and averages correctness over the same ten seeds as Table 1. Borda-JC uses the same candidate samples and judge evidence. This analysis uses full-pool Pass@1 only to define the bins, not as an input to any aggregation method.

Table 12: **MathArena-Full performance stratified by problem difficulty.** Bin 1 contains the five easiest questions from each benchmark and bin 6 the five hardest, as ranked by full-pool Pass@1; each row contains 20 questions in total. Δ_{WSC} and Δ_{Borda} denote the accuracy of JC minus WSC and Borda-JC, respectively, in percentage points. Boldface marks a strict row maximum.

ρ	Difficulty bin	Full-pool Pass@1	WSC	Borda-JC	JC	Δ_{WSC}	Δ_{Borda}
10%	1 (easiest)	97.1	100.0	100.0	100.0	+0.0	+0.0
	2	90.5	100.0	100.0	100.0	+0.0	+0.0
	3	83.7	100.0	100.0	100.0	+0.0	+0.0
	4	77.6	99.5	99.5	99.5	+0.0	+0.0
	5	68.0	99.5	99.5	98.5	−1.0	−1.0
	6 (hardest)	35.8	78.0	**81.5**	80.5	+2.5	−1.0
20%	1 (easiest)	97.1	100.0	100.0	100.0	+0.0	+0.0
	2	90.5	100.0	100.0	100.0	+0.0	+0.0
	3	83.7	100.0	100.0	100.0	+0.0	+0.0
	4	77.6	99.0	99.0	**100.0**	+1.0	+1.0
	5	68.0	100.0	100.0	100.0	+0.0	+0.0
	6 (hardest)	35.8	80.0	82.0	**84.0**	+4.0	+2.0
50%	1 (easiest)	97.1	100.0	100.0	100.0	+0.0	+0.0
	2	90.5	100.0	100.0	100.0	+0.0	+0.0
	3	83.7	100.0	100.0	100.0	+0.0	+0.0
	4	77.6	100.0	100.0	100.0	+0.0	+0.0
	5	68.0	100.0	100.0	100.0	+0.0	+0.0
	6 (hardest)	35.8	81.5	82.0	**84.5**	+3.0	+2.5

The first five bins are largely saturated: JC and WSC differ by at most one percentage point, and neither method has a consistent advantage. In the hardest bin, JC improves over WSC by 2.5, 4.0, and 3.0 points for $\rho = 10\%$, 20%, and 50%, respectively. The comparison with Borda-JC is more mixed at $\rho = 10\%$, where Borda-JC is one point higher, but JC leads by 2.0 and 2.5 points in the hardest bin at the two larger sampling ratios. The evidence therefore supports a difficulty-dependent benefit rather than uniform improvement across questions.

D.3 Alternative designs of interaction matrix J

Soft Borda based kernel. Borda-JC selects the answer maximizing this group-balanced Borda score plus $\mu \sum_{i \in \mathcal{I}_k} \mathbf{h}_i$, with the same $\mu = 0.5$ used by JC. It uses the same pairwise and pointwise judgments as JC and requires no additional judge calls.

RBF kernel based interaction. We first compute a Gaussian radial basis function (RBF) kernel from the columns of $\mathbf{C}$. The bandwidth β is chosen as the median of the positive pairwise Euclidean distances:

$$[\mathbf{K}_{\mathrm{RBF}}]_{uv} = \exp\left(-\frac{\|\mathbf{C}_{:,u} - \mathbf{C}_{:,v}\|_2^2}{2\beta^2}\right), \qquad \beta = \mathrm{median}\{\|\mathbf{C}_{:,u} - \mathbf{C}_{:,v}\|_2 : u < v\} > 0.$$

We apply kernel centering with $\mathbf{H} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$, followed by spectral normalization:

$$\widetilde{\mathbf{K}}_{\mathrm{RBF}} = \frac{\mathbf{H}\mathbf{K}_{\mathrm{RBF}}\mathbf{H}}{\lambda_{\max}(\mathbf{H}\mathbf{K}_{\mathrm{RBF}}\mathbf{H})}.$$

The corresponding interaction matrix is

$$\mathbf{J}_{\mathrm{RBF}} = \mathbf{C}\big[(1-\eta)\mathbf{I} + \eta\widetilde{\mathbf{K}}_{\mathrm{RBF}}\big]\mathbf{C}^\top, \qquad \eta \in [0,1]. \tag{8}$$

At $\eta = 0$, the construction reduces to the Gram matrix $\mathbf{C}\mathbf{C}^\top$. For $\eta > 0$, the RBF kernel additionally captures interactions between different columns of $\mathbf{C}$, weighted by the similarity of their comparative profiles. Kernel centering removes the common similarity component, while spectral normalization

ensures that the kernel has the same operator norm as the identity. The parameter η controls the balance between the Gram and RBF terms.

For both MathArena-Full and lower-tier AIME, we use $\mu = 0.5$ and $\eta = 0.2$. The RBF-JC results are included in Tables 13 and 14. On MathArena-Full, RBF-JC achieves an average accuracy of 97.11%, compared with 97.06% for JC. On lower-tier AIME, RBF-JC achieves 62.83%, compared with 61.89% for JC and Borda-JC, and obtains the best or tied-best performance in five of the six settings.

MathArena-Full Results (Table 1) with Standard Deviation Table 13 provides the complete version of Table 1, including sample standard deviations over 10 seeds and the RBF-JC variant described in Appendix D.3.

Table 13: **Full MathArena-Full results corresponding to Table 1.** Entries report mean accuracy $\pm$ sample standard deviation over 10 seeds; RBF-JC is included as an additional variant.

Dataset	ρ	Pass@1	BoN	KT	SC	WSC	Copeland	Elo	BT	Borda-JC	RBF-JC	JC
AIME 2025	10%	70.3 ± 0.9	87.6 ± 2.3	96.0 ± 2.1	94.0 ± 2.1	95.3 ± 1.9	92.0 ± 1.9	83.3 ± 5.2	89.7 ± 4.6	$\mathbf{96.3} \pm 2.9$	95.7 ± 3.2	95.7 ± 3.2
	20%	69.6 ± 0.7	88.4 ± 1.0	94.7 ± 1.7	93.2 ± 1.2	$\mathbf{96.3} \pm 1.1$	88.3 ± 0.0	78.7 ± 6.5	90.0 ± 1.6	$\mathbf{96.3} \pm 1.1$	96.0 ± 2.1	95.7 ± 1.6
	50%	69.7 ± 0.4	88.4 ± 0.6	96.0 ± 1.4	94.5 ± 1.6	$\mathbf{96.7} \pm 0.0$	88.3 ± 0.0	74.3 ± 4.5	90.0 ± 0.0	$\mathbf{96.7} \pm 0.0$	$\mathbf{96.7} \pm 0.0$	$\mathbf{96.7} \pm 0.0$
BRUMO 2025	10%	85.8 ± 1.6	94.5 ± 1.4	98.0 ± 2.3	97.3 ± 2.4	99.3 ± 1.4	95.1 ± 3.5	90.3 ± 4.3	94.3 ± 3.5	$\mathbf{100.0} \pm 0.0$	$\mathbf{100.0} \pm 0.0$	$\mathbf{100.0} \pm 0.0$
	20%	86.2 ± 0.7	94.3 ± 1.2	98.7 ± 2.3	98.5 ± 1.7	99.7 ± 1.1	95.0 ± 3.3	90.7 ± 4.4	95.0 ± 2.4	$\mathbf{100.0} \pm 0.0$	$\mathbf{100.0} \pm 0.0$	$\mathbf{100.0} \pm 0.0$
	50%	86.0 ± 0.4	94.3 ± 0.6	99.7 ± 1.1	98.2 ± 1.7	$\mathbf{100.0} \pm 0.0$	90.1 ± 2.0	82.0 ± 3.9	90.3 ± 2.9	$\mathbf{100.0} \pm 0.0$	$\mathbf{100.0} \pm 0.0$	$\mathbf{100.0} \pm 0.0$
HMMT 2025 (Feb)	10%	58.4 ± 1.6	83.9 ± 1.9	90.7 ± 7.5	92.7 ± 2.1	96.2 ± 2.9	80.7 ± 3.4	78.7 ± 5.7	86.3 ± 2.5	96.3 ± 2.9	$\mathbf{97.0} \pm 1.9$	$\mathbf{97.0} \pm 1.9$
	20%	58.1 ± 1.1	82.9 ± 0.8	94.3 ± 2.7	94.5 ± 1.8	96.0 ± 2.6	80.1 ± 2.2	81.7 ± 3.9	84.7 ± 1.7	96.7 ± 2.7	$\mathbf{98.7} \pm 1.7$	$\mathbf{98.7} \pm 1.7$
	50%	58.3 ± 0.6	83.4 ± 0.8	96.0 ± 2.6	94.0 ± 1.4	96.7 ± 1.6	78.3 ± 0.2	75.3 ± 6.5	83.7 ± 1.1	97.0 ± 1.9	98.3 ± 1.8	$\mathbf{98.7} \pm 1.7$
HMMT 2025 (Nov)	10%	87.4 ± 1.8	88.5 ± 3.1	90.0 ± 4.2	93.5 ± 1.5	93.8 ± 2.4	84.1 ± 6.4	84.7 ± 7.2	83.7 ± 6.2	$\mathbf{94.3} \pm 2.7$	93.7 ± 2.5	93.0 ± 2.9
	20%	87.8 ± 1.1	89.4 ± 0.9	92.7 ± 3.1	93.5 ± 0.5	94.0 ± 1.4	80.0 ± 6.3	79.7 ± 4.6	78.3 ± 6.3	94.3 ± 1.6	$\mathbf{95.0} \pm 2.4$	$\mathbf{95.0} \pm 2.4$
	50%	87.7 ± 0.5	89.6 ± 0.7	94.0 ± 2.1	93.3 ± 0.0	$\mathbf{94.3} \pm 1.6$	68.3 ± 3.5	69.3 ± 5.8	68.0 ± 3.2	$\mathbf{94.3} \pm 1.6$	$\mathbf{94.3} \pm 1.6$	$\mathbf{94.3} \pm 1.6$

Lower-Tier AIME Results (Table 3) with Standard Deviation Table 14 provides the complete version of Table 3, including the standard-deviation statistics under the Table 3 reporting convention and the additional RBF-JC variant.

Table 14: **Full lower-tier AIME results corresponding to Table 3.** Entries report mean accuracy $\pm$ standard deviation across 10 random seeds, with each seed's accuracy computed over all 30 problems. RBF-JC is included as an additional variant. The final row reports the average over R without aggregating the per-setting standard deviations.

R	Pass@1	BoN	KT	SC	WSC	Copeland	Elo	BT	Borda-JC	RBF-JC	JC
3	11.2 ± 2.2	23.7 ± 4.3	9.3 ± 1.4	19.6 ± 3.8	23.5 ± 4.2	26.7 ± 4.4	24.3 ± 4.3	$\mathbf{27.0} \pm 4.4$	$\mathbf{27.0} \pm 4.4$	$\mathbf{27.0} \pm 4.4$	$\mathbf{27.0} \pm 4.4$
5	16.0 ± 2.6	31.8 ± 4.7	13.6 ± 0.9	29.5 ± 4.5	31.8 ± 4.6	34.3 ± 4.8	32.0 ± 4.7	$\mathbf{36.3} \pm 4.8$	$\mathbf{36.3} \pm 4.8$	$\mathbf{36.3} \pm 4.8$	$\mathbf{36.3} \pm 4.8$
10	26.9 ± 3.2	50.0 ± 5.0	40.9 ± 3.7	43.0 ± 4.9	51.6 ± 4.9	59.3 ± 4.9	46.3 ± 5.0	$\mathbf{60.0} \pm 4.9$	58.0 ± 4.9	$\mathbf{60.0} \pm 4.9$	59.7 ± 4.9
15	34.3 ± 3.2	63.1 ± 4.8	57.9 ± 4.4	54.7 ± 4.9	68.7 ± 4.6	69.2 ± 4.4	48.7 ± 5.0	76.3 ± 4.3	74.0 ± 4.4	$\mathbf{77.7} \pm 4.2$	76.0 ± 4.3
20	40.9 ± 3.2	67.3 ± 4.7	69.4 ± 3.7	71.5 ± 4.4	77.7 ± 4.2	70.8 ± 4.4	52.7 ± 5.0	70.7 ± 4.6	84.3 ± 3.6	$\mathbf{84.7} \pm 3.6$	82.3 ± 3.8
30	52.1 ± 3.0	76.5 ± 4.2	83.6 ± 2.6	83.3 ± 3.7	88.2 ± 3.2	63.7 ± 4.5	44.7 ± 5.0	60.7 ± 4.9	$\mathbf{91.7} \pm 2.8$	91.3 ± 2.8	90.0 ± 3.0
Average	30.23	52.07	45.78	50.27	56.91	54.00	41.44	55.17	61.89	**62.83**	61.89

Ablation of the independent evaluation term. We compare Gram-JC and Borda-JC with and without $\mathbf{h}$ on MathArena-Full and in a separate lower-tier AIME experiment. The latter uses the weakest $R \in \{3, 5, 10, 15, 20, 30\}$ generators with 50% trace sampling per problem. Within each paired ablation, the sampled traces, pairwise judgments, and interaction construction are held fixed, and the "without $\mathbf{h}$" variant only removes $\mu\langle \mathbf{h}, \mathbf{x} \rangle$. The full variants use $\mu = 0.5$. Table 15 reports averages over the 12 cells of Table 1 and the six values of R in the lower-tier AIME experiment.

Table 15: **Effect of the independent evaluation term on comparative aggregation.** Entries report mean accuracy (%); no hyperparameter is retuned after removing $\mathbf{h}$. $\Delta_{\mathbf{h}}$ is the signed change in percentage points from adding $\mathbf{h}$, computed before rounding. Bold denotes the best or tied-best construction in each accuracy column.

	Table 1			Lower-tier AIME		
Comparative component	w/o $\mathbf{h}$	w/ $\mathbf{h}$	$\Delta_{\mathbf{h}}$	w/o $\mathbf{h}$	w/ $\mathbf{h}$	$\Delta_{\mathbf{h}}$
Gram-JC	84.66	**97.06**	+12.40	64.29	**64.83**	+0.54
Borda-JC	**86.17**	96.86	+10.69	**65.96**	62.83	−3.12

Adding $\mathbf{h}$ improves the average accuracy of both constructions on MathArena-Full, by 12.40 and 10.69 percentage points for Gram-JC and Borda-JC, respectively. In the separate lower-tier AIME experiment, adding $\mathbf{h}$ improves Gram-JC by 0.54 points but reduces Borda-JC by 3.12 points. This indicates that the fixed μ need not transfer across comparative scores with different numerical scales.

D.4 Sensitivity to Pair Order and Prompt Design

Pair order. For each pairwise comparison, we query the judge twice with the positions of the two traces swapped. Let s_{ij} be the probability returned when y_i is presented first and s_{ji} the probability returned when y_j is presented first. We use the bidirectionally normalized probabilities

$$P_\theta(y_i \succeq y_j) = \frac{\exp(s_{ij})}{\exp(s_{ij}) + \exp(s_{ji})}, \qquad P_\theta(y_j \succeq y_i) = \frac{\exp(s_{ji})}{\exp(s_{ij}) + \exp(s_{ji})}.$$

To evaluate sensitivity to presentation order, we evaluate aggregation in the setting of Table 1 using either both directions, only the forward judgment, or only the reverse judgment. For each single-direction condition, the unqueried opposite score is inferred as the complement of the retained probability before applying the same normalization. All conditions use identical sampled traces, independent scores, seeds, and JC objective, and require no additional judge calls.

Table 16: **Sensitivity of JC to pair presentation order on MathArena-Full.** Entries report mean accuracy $\pm$ sample standard deviation. The bidirectional column reproduces the JC entries in Table 1.

Dataset	ρ	Bidirectional	Forward only	Reverse only
AIME 2025	10%	95.7 ± 3.2	93.3 ± 1.6	97.3 ± 3.4
	20%	95.7 ± 1.6	93.7 ± 1.1	98.7 ± 1.7
	50%	96.7 ± 0.0	96.3 ± 1.1	96.7 ± 0.0
BRUMO 2025	10%	100.0 ± 0.0	99.7 ± 1.1	100.0 ± 0.0
	20%	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
	50%	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
HMMT 2025 (Feb)	10%	97.0 ± 1.9	95.3 ± 2.3	96.0 ± 3.1
	20%	98.7 ± 1.7	98.7 ± 1.7	98.7 ± 1.7
	50%	98.7 ± 1.7	98.0 ± 2.3	98.3 ± 1.8
HMMT 2025 (Nov)	10%	93.0 ± 2.9	91.0 ± 3.2	93.7 ± 3.7
	20%	95.0 ± 2.4	93.0 ± 1.9	96.7 ± 2.2
	50%	94.3 ± 1.6	93.7 ± 1.1	95.3 ± 1.7

Bidirectional querying obtains an accuracy of 97.1%, averaged uniformly over the 12 combinations. The corresponding accuracy averaged over the two single-order conditions is 96.8%, a difference of 0.22 percentage points. This indicates that the aggregate result is similar when averaged over the two possible presentation orders in this setting.

Prompt wording and output format. We additionally compare three pairwise-judge prompts on lower-tier AIME traces: the original probability prompt, a paraphrased probability prompt, and a direct binary prompt that requests only which response is better. We use a fixed comparison set of 7,840 sampled trace pairs spanning 784 answer pairs, each evaluated in both presentation orders for all three prompts. We consider $R \in \{3, 5, 10, 15, 20, 30\}$ weakest generators, with 50%-per-problem

sampling, ten repeats, and $\mu = 0.5$. The judge model and settings, trace pairs and presentation orders, sampled traces, independent scores, JC objective, and tie handling are held fixed across prompts. The original probability template is given in Appendix F.1.2. The complete paraphrased probability and direct binary templates are given in Appendix F.1.3 and F.1.4, respectively.

Table 17: **Sensitivity of JC to pairwise-judge prompt design on lower-tier AIME traces.** Entries report mean accuracy (%) $\pm$ standard deviation (in percentage points) across 10 random seeds, with each seed's accuracy computed over all 30 problems. All three prompts use identical trace pairs and aggregation samples.

R	Original probability	Paraphrased probability	Direct binary
3	26.3 ± 4.4	26.3 ± 4.4	26.3 ± 4.4
5	36.3 ± 4.8	36.3 ± 4.8	36.3 ± 4.8
10	61.3 ± 4.9	63.3 ± 4.8	63.7 ± 4.8
15	80.0 ± 4.0	78.0 ± 4.1	85.3 ± 3.5
20	89.7 ± 3.0	88.3 ± 3.2	89.0 ± 3.1
30	95.3 ± 2.1	96.0 ± 2.0	94.5 ± 2.3

The accuracies averaged over the six values of R are 64.8%, 64.7%, and 65.9% for the original probability, paraphrased probability, and direct binary prompts, respectively. The overall accuracies are similar across the three prompts, although individual settings can differ more substantially; for example, at $R = 15$, direct binary judging improves over the original probability prompt by 5.3 percentage points. These results indicate comparable average performance for the three tested prompts under this comparison protocol.

E Derivation of the Gram–Borda Decomposition

Fix an answer group $\mathcal{I}_k$ and let U be uniformly distributed over its trace indices. Write $Z_j = \sqrt{p_{Uj}}$, where $p_{ij} = p_\theta(y_i \succeq y_j)$. For $\mathbf{x} = \mathbf{1}_{\mathcal{I}_k}$, the definitions of $\mathbf{C}$ and $\mathbf{J}_{\mathrm{Borda}}$ give

$$\mathbf{x}^\top \mathbf{J}\mathbf{x} = \|\mathbf{C}^\top \mathbf{x}\|_2^2 = \sum_{j=1}^{N} \frac{1}{n_j} \big(\mathbb{E}[Z_j]\big)^2,$$

$$\mathbf{x}^\top \mathbf{J}_{\mathrm{Borda}} \mathbf{x} = \sum_{j=1}^{N} \frac{1}{n_j} \mathbb{E}[Z_j^2].$$

Subtracting the two expressions and using $\mathrm{Var}(Z_j) = \mathbb{E}[Z_j^2] - (\mathbb{E}[Z_j])^2$ proves (7). The expectations and variances are over U only, with the preference values held fixed; no answer-level homogeneity assumption is needed.

F Prompt Templates

F.1 Prompts Used in Math Reasoning Tasks

F.1.1 Ising-h Prompt

The prompt to query $\mathbf{h}$ in math reasoning tasks consists of three parts as follows.

user

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

assistant

{response (cot+answer)}

user

Please evaluate the above answer based on the following criteria:

1. Is the answer correct?
2. Is the reasoning process correct?

Please choose an evaluation score among 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0. Please only output the evaluation score.

F.1.2 Ising-J Prompt

user

Suppose there are two responses to the same question. Please output the probability that Response 1 is a better answer than Response 2.

Question
{question}

Response 1
{response1 (cot+answer)}

Response 2
{response2 (cot+answer)}

Instruction
Now, please output the probability (a real number between 0 and 1) that Response 1 is a better answer than Response 2. Please only output the number.

F.1.3 Paraphrased Probability Prompt

The following prompt is sent as a single `user` message with no additional system prompt. Its output is constrained to a single number in $[0, 1]$.

user

You will be given a question together with two candidate responses. Estimate the probability that Response 1 is the better answer.
Question
{question}
Candidate Response 1
{response1}
Candidate Response 2
{response2}
Output Instruction
Return a single number between 0 and 1 representing the probability that Response 1 is better than Response 2. Output only the number, with no explanation or additional text.

F.1.4 Direct Binary Prompt

The following prompt is sent as a single `user` message with no additional system prompt. Its output is constrained to exactly `A` or `B`.

user

Suppose there are two responses to the same question. Choose which response is the better answer.
Question
{question}
Response A
{response_a}
Response B
{response_b}
Instruction
Which response is better? Output exactly one uppercase letter: A or B. Do not output a score, probability, explanation, or any other text.

F.1.5 Generation Prompt

user

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

F.2 Prompt Used in Code Reasoning Tasks

F.2.1 Ising-h Prompt

The prompt to query $\mathbf{h}$ in code reasoning tasks consists of three parts as follows.

user

Given the following Python function and input, predict the output.
Function: {code_str}
Input: {input_str}
Please think step by step after "Reasoning:`\n\n`" and then leave the output after "Output:`\n\n`". Note the output should be a python object and please ignore markdown format.{question}

assistant

{response}

user

Please evaluate the above answer based on the following criteria:

1. Is the answer correct?
2. Is the reasoning process correct?

Please choose an evaluation score among 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0.
Please only output the evaluation score.

F.2.2 Ising-J Prompt

user

Suppose there are two responses to the same Python function and input. Please output the probability that Response 1 is a better answer than Response 2.

Python function and input

Function:
{code_str}

Input:
{input_str}

Response 1
{content1}

Response 2
{content2}

Instruction

Now, please output the probability (a real number between 0 and 1) that Response 1 is a better answer than Response 2. Please only output the number.

F.2.3 Generation Prompts

user

Given the following Python function and input, predict the output.
Function: {code_str}
Input: {input_str}
Please think step by step after "Reasoning:\n\n" and then leave the output after "Output:\n\n". Note the output should be a python object and please ignore markdown format.{question}

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our framework is established in §3 while our empirical results are presented in §4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have presented our assumption in §3 and pointed out the number of random trials to reduce the influence of stochasticity in §4. The limitation of the paper is discussed in §5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The exact Gram–Borda decomposition is stated in §3.2 and derived in §E.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We gave prompts in §F and hyperparameters in §B. Other setups can be found in §4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The dataset used in this paper (AIME'25, HMMT'25-Feb, HMMT'25-Nov, BRUMO'25, CRUXEval) can be found on HuggingFace. The attached supplemental matrial provide codes to reproduce results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We gave instructions and details in §4 and §B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All experiments are repeated for 10 different seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Listed in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: There is no societal impact or potential harmful consequences that we can think of.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper focus on solving reasoning on math or coding problem and does not have societal impacts that we can think of.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We release no new models, datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited paper of models, datasets in §4.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects. Notably, the crowdsourced dataset, namely MathArena traces, are generated by diverse LLMs, not human.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are only used in polishing the paper. The core methodology, scientific rigor, or originality of the research are not influenced by LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.