DON'T PASS@k: A BAYESIAN FRAMEWORK FOR LARGE LANGUAGE MODEL EVALUATION

Mohsen Hariri,[*] **Amirhossein Samandar,**[†] **Michael Hinczewski,**[†] **Vipin Chaudhary**[*]
[*]Department of Computer and Data Sciences, Case Western Reserve University, Cleveland, OH, USA
[†]Department of Physics, Case Western Reserve University, Cleveland, OH, USA
{mohsen.hariri, axs2935, mxh605, vipin}@case.edu

ABSTRACT

Pass@k is widely used to report performance for LLM reasoning, but it often yields unstable, misleading rankings, especially when the number of trials (samples) is limited and compute is constrained. We present a principled Bayesian evaluation framework that replaces Pass@k and average accuracy over N trials (avg@N) with posterior estimates of a model's underlying success probability and credible intervals, yielding stable rankings and a transparent decision rule for differences. Evaluation outcomes are modeled as categorical (not just 0/1) with a Dirichlet prior, giving closed-form expressions for the posterior mean and uncertainty of any weighted rubric and enabling the use of prior evidence when appropriate. Theoretically, under a uniform prior, the Bayesian posterior mean is order-equivalent to average accuracy (Pass@1), explaining its empirical robustness while adding principled uncertainty. Empirically, in simulations with known ground-truth success rates and on AIME'24/'25, HMMT'25, and BrUMO'25, the Bayesian/avg procedure achieves faster convergence and greater rank stability than Pass@k and recent variants, enabling reliable comparisons at far smaller sample counts. The framework clarifies when observed gaps are statistically meaningful (non-overlapping credible intervals) versus noise, and it naturally extends to graded, rubric-based evaluations. Together, these results recommend replacing Pass@k for LLM evaluation and ranking with a posterior-based, compute-efficient protocol that unifies binary and non-binary evaluation while making uncertainty explicit. Source code is available at https://mohsenhariri.github.io/bayes-kit.

1 INTRODUCTION

Large language models (LLMs) have moved rapidly from research artifacts to everyday infrastructure (1; 2). Students use them for homework and exam preparation; developers rely on them for code synthesis and refactoring (3); analysts and clinicians use them for decision support; and agents built atop LLMs are increasingly embedded in workflows across industry and government. This demand has catalyzed unprecedented investment: specialized chips, datacenters, and startups dedicated to LLM training, serving, and tooling (4). As deployment accelerates, trust, oversight, and comparability become central: *how we evaluate LLMs* directly shapes which models are adopted, what progress is declared, and how resources are allocated (5; 6; 7; 8; 9; 10; 11).

Evaluation, however, remains the weakest link in the LLM pipeline. Alongside advances in model efficiency and compression(12; 13; 14; 15; 16; 17; 18), training and fine-tuning (PEFT/LoRA, RL-from-human-feedback) (19; 20; 11), and inference/decoding (sampling strategies, caching, efficient attention) (21; 22), the community still leans on simple, yet flawed, success rates and Pass@k-style metrics to summarize capabilities (23). These practices are convenient but fragile. On small or costly benchmarks (e.g., math reasoning sets with only tens of problems such as AIME) (24), Pass@k or single-run accuracy often produce unstable rankings (25; 26), are sensitive to decoding choices and seed effects (27; 25), and provide little guidance on whether observed gaps are meaningful or mere noise (28; 29). Averaging across multiple runs ("avg@N") helps but is compute-hungry (30), offers no unified way to handle graded/rubric outcomes, and lacks a principled decision rule for significance (28; 31; 32).

This paper takes a different approach: we treat evaluation itself as a statistical inference problem. We introduce a *posterior-based* framework that replaces Pass@k and avg@N with estimates of a model's underlying success probabilities and associated uncertainty (33). Outcomes are modeled as *categorical* (34) rather than purely binary: each item can yield correct, partially correct, formatting-error, refusal, or rubric-defined levels. A Dirichlet prior over these categories yields closed-form posterior means and credible intervals for any *weighted rubric*, allowing the evaluator to report both a point estimate and principled uncertainty with negligible overhead. In the binary special case under

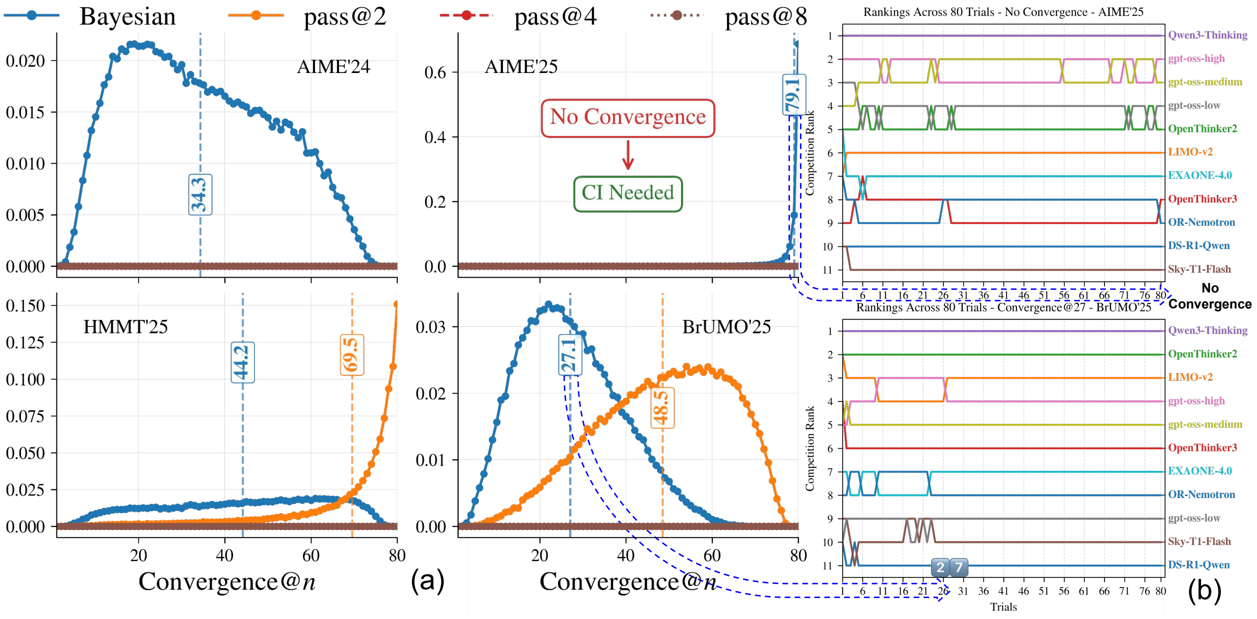

Figure 1: a) Probability mass functions (PMFs) of convergence@n, the number of trials n above which a ranking of LLM models consistently matches the ranking using $N_{\max} = 80$ trials. Eleven LLM models (listed on the right) and four math-reasoning datasets are used—AIME'24, AIME'25, HMMT'25, and BrUMO'25—comparing Pass@2/4/8 against our Bayesian posterior evaluation (Bayes@N). Each PMF is estimated by bootstrapping with 10^5 samples over the $N_{\max}$ trials; vertical lines indicate the mean of each convergence distribution. On AIME'24/'25, the Pass family frequently *fails to converge*, whereas Bayes@N converges. On HMMT and BrUMO, Pass methods converge more slowly (mean required trials ≈ 69.5 and ≈ 48.5) than Bayes@N (≈ 44.2 and ≈ 27.1), respectively. Right: Example competition-style ranking from a single bootstrap replicate, highlighting the mean convergence for AIME'25 and BrUMO'25. Per task rankings, including worst-case replicates, are in Section 3.3 (Fig. 5).

a uniform prior, its posterior mean is order-equivalent to average accuracy, explaining the empirical robustness of avg@N while making uncertainty explicit.

The framework addresses *four* persistent pain points. ❶ *Convergence*: as shown in Fig. 1, we ideally want methods that can converge to the true underlying ranking with the smallest number of trials, but different approaches can have significantly different convergence speeds. ❷ *Credible intervals*: a simple, transparent rule—**do not declare a winner when intervals overlap**—reduces leaderboard churn and over-interpretation of tiny gaps by introducing a compute-efficient confidence interval (CI). Updates are analytic; one can monitor interval widths online, and allocate additional trials only when needed (no Monte Carlo/bootstrap simulations are required for CI estimation). ❸ *Categorical evaluation*: our approach unifies binary and non-binary evaluation. Graded rubrics are natural in this framework, so one can evaluate step-by-step reasoning, partial credit, or judge categories without ad hoc aggregation. ❹ *Prior information*: we can incorporate prior evidence when appropriate (e.g., reuse of stable rubric distributions across closely related tasks or versions).

We validate the approach in two settings: In controlled simulations with known ground-truth success rates, the posterior procedure converges to correct rankings with fewer samples than Pass@k and recent variants, and it flags when ties are statistically unresolved. On real math-reasoning benchmarks (AIME'24/'25 (35; 36), HMMT'25 (37), and BrUMO'25 (38)-derived sets), we observe the same pattern: the posterior method achieves greater rank stability at far smaller sample counts than Pass@k, while clarifying when differences are meaningful versus noise. Practically, this yields a computationally efficient protocol that is easy to implement and audit.

We summarize our contributions as follows:

- **A unified Bayesian evaluation framework.** We model per-item outcomes as categorical with a Dirichlet prior, yielding closed-form posterior means and credible intervals for *any* weighted rubric, with binary evaluation as a special case. This unifies 0/1 and graded evaluations and supports reuse of prior evidence when justified.

- **A compute-efficient, interval-aware protocol.** We provide a simple recipe: report posterior means with credible intervals; only declare differences when intervals do not overlap; adaptively allocate additional samples until intervals meet pre-specified widths. This protocol naturally supports sequential/online evaluation.
- **Empirical evidence on simulations and math benchmarks.** On synthetic data with known ground truth and on AIME'24/'25, HMMT'25, and BrUMO'25 datasets, our method achieves faster convergence and greater rank stability than Pass@k and recent variants, enabling reliable comparisons with far fewer samples.

2 Bayesian Framework for Evaluating LLM Performance

2.1 Background: The Pass@k Metric and Its Limitations

Evaluation metrics for LLMs aim to quantify performance on tasks like reasoning or programming, but they often struggle to provide reliable relative rankings across models. Pass@k, for instance, estimates the probability of at least one correct answer within k model attempts (see Section E for details). While convenient, this metric exhibits high variance (39), particularly when k approaches the total number of trials, N, resulting in unstable rankings (40). Small fluctuations in correctness can distort comparisons, particularly in benchmarks with few problems or limited computational resources, raising doubts about its suitability for differentiating model capabilities. If a metric cannot consistently distinguish stronger models from weaker ones, its value as a benchmarking tool is undermined (26).

Estimating uncertainty in Pass@k scores is also challenging, as it lacks closed-form expressions for variance, relying instead on computationally intensive approximations like bootstrapping. A truly effective metric should yield reliable performance rankings with a minimal number of trials, prioritizing both accuracy and efficiency in resource-constrained environments. To address these limitations, we propose a Bayesian evaluation framework that provides more stable estimates of performance, incorporates uncertainty, and facilitates robust relative comparisons across models (33; 41; 42).

2.2 Results Matrix

Consider a results matrix R for an LLM evaluated on a test set comprising M questions. Due to the stochastic nature of LLM sampling, responses may vary across independent trials, so we run the LLM N times per question. The outcomes are captured in the $M \times N$ matrix R, where element $R_{\alpha i}$ represents the score in the ith trial for the αth question. This score is an integer ranging from 0 to a maximum value C, reflecting a rating system with $C + 1$ categories. In the binary case ($C = 1$), 0 indicates an incorrect answer and 1 a correct one, though we accommodate more nuanced rubrics generally.

2.3 Weighted Performance Metric

For the αth question, $\alpha = 1, \ldots, M$, there is an underlying probability $\pi_{\alpha k}$ that the LLM's answer falls in the kth category. We denote $\boldsymbol{\pi}_\alpha$ as the $(C+1)$-dimensional vector with elements $\pi_{\alpha k}$, $k = 0, \ldots, C$. If all $\boldsymbol{\pi}_\alpha$ were known, we could calculate a desired performance metric $\bar{\pi}$ as a weighted average over these probabilities:

$$\bar{\pi} = \frac{1}{M} \sum_{\alpha=1}^{M} \boldsymbol{w} \cdot \boldsymbol{\pi}_\alpha = \frac{1}{M} \sum_{\alpha=1}^{M} \sum_{k=0}^{C} w_k \pi_{\alpha k}, \tag{1}$$

where $\boldsymbol{w}$ is a $(C+1)$-dimensional vector of constant weights. For example, if $w_k = k$, then $\bar{\pi}$ represents the average category label. In the case where $C = 1$, this average corresponds to the mean probability of a correct answer over the entire test set. However, we allow for a general choice of $\boldsymbol{w}$ to accommodate a wide range of possible metrics.

2.4 Bayesian Estimator and Uncertainty for the Performance Metric

In principle, we could estimate $\boldsymbol{\pi}_\alpha$ by running an arbitrarily large number of trials with the LLM, yielding an accurate estimate of $\bar{\pi}$. However, we are typically constrained to small N due to limited computational resources. Our goal is to develop a Bayesian approach to estimate $\bar{\pi}$ and its associated uncertainty given a finite N. The first step is to construct $\mathcal{P}(\boldsymbol{\pi}_\alpha | \boldsymbol{R}_\alpha)$, the posterior probability of $\boldsymbol{\pi}_\alpha$ given the αth row of the matrix R, denoted $\boldsymbol{R}_\alpha$. This posterior depends on the data in $\boldsymbol{R}_\alpha$ and a chosen prior distribution $\mathcal{P}(\boldsymbol{\pi}_\alpha)$ for the unknown underlying probability vector $\boldsymbol{\pi}_\alpha$. The prior could be uniform (assuming no prior information) or incorporate previously gathered evidence about the LLM's performance. The Bayesian framework focuses on two quantities: the first is the mean of $\bar{\pi}$ over the joint posterior for all questions, which we denote as $\mu(R)$. This is a Bayesian optimal estimator, minimizing the quadratic loss function $\mathcal{L}(\bar{\pi}^{\text{est}}) = \mathbb{E}_{R,\boldsymbol{\pi}_\alpha}(\bar{\pi}^{\text{est}}(R) - \bar{\pi})^2$ over all possible estimators $\bar{\pi}^{\text{est}}(R)$, where the expectation value is over all possible $\boldsymbol{\pi}_\alpha$ and realizations of R (43). The second quantity is the variance $\sigma^2(R)$, which quantifies the uncertainty of the μ estimate. Both $\mu(R)$ and $\sigma^2(R)$ have exact closed-form expressions, derived in Appendix A, and can be simply calculated for any R using Algorithm 1.

Algorithm 1 LLM performance evaluation using the Bayes@N framework.

```
function EVALUATEPERFORMANCE($R$, $[R^0]$, $\boldsymbol{w}$)
    input: $M \times N$ matrix $R$ of results, with each element $R_{\alpha i} = 0, \ldots, C$
           weight vector $\boldsymbol{w} = (w_0, \ldots, w_C)$ defining performance metric $\bar{\pi}$
    optional input: $M \times D$ matrix $R^0$ of results for prior; otherwise $D = 0$
    output: performance metric estimate $\mu$ and associated uncertainty $\sigma$
    $T = 1 + C + D + N$
    for $\alpha = 1$ to $M$ do                                   ▷ Tally results in $R$ and $R^0$
        for $k = 0$ to $C$ do
            $n_{\alpha k} = \sum_{i=1}^{N} \delta_{k,R_{\alpha i}}$
            $n^0_{\alpha k} = 1 + \sum_{i=1}^{D} \delta_{k,R^0_{\alpha i}}$
            $\nu_{\alpha k} = n^0_{\alpha k} + n_{\alpha k}$
        end for
    end for
    $\mu = w_0 + \frac{1}{MT} \sum_{\alpha=1}^{M} \sum_{j=0}^{C} \nu_{\alpha j}(w_j - w_0)$
    $\sigma = \left[ \frac{1}{M^2(T+1)} \sum_{\alpha=1}^{M} \left\{ \sum_{j=0}^{C} \frac{\nu_{\alpha j}}{T}(w_j - w_0)^2 - \left( \sum_{j=0}^{C} \frac{\nu_{\alpha j}}{T}(w_j - w_0) \right)^2 \right\} \right]^{1/2}$
    return $\mu$, $\sigma$
end function
```

2.5 Using Uncertainty Estimates to Decide Significance of Performance Differences

In general, the expressions for $\mu(R)$ and $\sigma^2(R)$ are valid for any M and N, and do not rely on asymptotic arguments like the central limit theorem (CLT). However, there are useful simplifications that occur in specific limiting cases. For example as the size of the test set M becomes large, we can derive not just the moments of the posterior distribution for $\bar{\pi}$, but also its shape, which becomes approximately Gaussian: $\mathcal{P}(\bar{\pi}|R) \sim \mathcal{N}(\mu(R), \sigma^2(R))$. This allows us to assess whether two methods exhibit a statistically significant performance difference. Consider results matrices R and R' from two approaches, with corresponding means μ, μ' and standard deviations σ, σ'. The distribution of the performance difference $\Delta\bar{\pi} \equiv \bar{\pi} - \bar{\pi}'$ is a convolution of the individual posteriors, yielding another normal distribution: $\mathcal{P}(\Delta\bar{\pi}|R, R') \sim \mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2)$, where the mean of the difference is $\tilde{\mu} = \mu - \mu'$, and the standard deviation is $\tilde{\sigma} = \sqrt{\sigma^2 + (\sigma')^2}$. To determine our confidence in the ranking of the two methods, we need to determine the probability that $\text{sign}(\Delta\bar{\pi}) = \text{sign}(\mu - \mu')$. This can be done by calculating the absolute z-score, $z = |\mu - \mu'|/\sqrt{\sigma^2 + (\sigma')^2}$. The probability that the ranking based on μ and μ' is correct (the ranking confidence ρ) is given by $\rho = (1/2)(1 + \text{erf}(z/\sqrt{2}))$. For example $z = 1.645$ corresponds to $\rho = 0.95$.

2.6 Equivalence of Bayesian and Average Rankings for Uniform Prior

In the results below, we will denote ranking based on the Bayesian estimator μ with a uniform prior as Bayes@N. Because μ is related to a naive weighted average accuracy via a positive affine transformation, it turns out the ranking based on the average, denoted as avg@N, is identical to Bayes@N (for the detailed proof, see Appendix B). In the large-trial limit $N \to \infty$, the value of μ approaches the average, as expected, but the ranking equivalence holds at all finite N. This relationship also extends to uncertainty quantification, where the standard deviation of the average relates to the Bayesian standard deviation σ by a scaling factor, providing a concrete method to compute uncertainty in the average without relying on the Central Limit Theorem (CLT). This is particularly advantageous in small-sample regimes common in LLM evaluations, where CLT-based methods often underestimate uncertainty and produce invalid intervals (e.g., extending beyond [0,1] or collapsing to zero) (44). As highlighted by (44), Bayesian approaches with uniform priors (e.g., Beta(1,1) in the binary case) yield well-calibrated credible intervals even for datasets with fewer than a few hundred datapoints, outperforming CLT approximations in coverage and handling complex structures like clustered data.

2.7 Gold Standard for Ranking

Strictly speaking, the underlying true ranking of LLMs for a particular performance metric $\bar{\pi}$ is unknown, because it would require determining the infinite trial limit, $\bar{\pi} = \lim_{N\to\infty} \mu$, for each LLM. In practice, we have to settle for an approximation to $\bar{\pi}$, calculated at some large but finite value $N = N_{\max}$ (for example $N_{\max} = 80$ in our LLM experiments). Specifically we will use Bayes@$N_{\max}$—which is the same as the ranking based on avg@$N_{\max}$—as our "gold standard" or reference ranking. In other words, rankings using smaller N will be compared to this gold standard to assess their accuracy.

For this comparison, we employ Kendall's τ, a nonparametric correlation coefficient that measures ordinal agreement between two rankings by comparing the number of concordant and discordant pairs of models. The coefficient ranges from -1 (perfect inversion) to $+1$ (perfect agreement), with 0 indicating no association. We specifically use the τ_b variant, which properly accounts for ties in the rankings (e.g., the intentional tie in our simulation below), ensuring that equivalences do not artificially inflate the correlation. See Appendix F.1 for further discussion and formal definitions.

To validate our claims about the gold standard as Bayes@$N_{\max}$, specifically to determine which evaluation methods converge to the true ranking, we conduct a simulation using biased coins as a metaphor for LLMs. In this setup, we already know the underlying performance distribution (the success probabilities $\boldsymbol{\pi}_\alpha$ for each question), allowing us to establish a known ground truth $\bar{\pi}$. Briefly, a biased coin mimic of a stochastic LLM with $C = 1$ consists of M values $\pi_{\alpha 1}$, representing the chances of answering each question α correctly. These probabilities are drawn from Beta distributions with shape parameters Beta$(i, 18 - i)$ for i ranging from 4 to 13, with the set for $i = 7$ duplicated to create identical performances for two models, and the set for $i = 13$ drawn independently for the eleventh model. We generate 11 sets of these 30 probabilities, with $\bar{\pi}$ values of $[0.2332, 0.2545, 0.3604, 0.3642, 0.3642, 0.4466, 0.5418, 0.5276, 0.608, 0.6213, 0.7327]$, representing different LLMs (note the tie at 0.3642 to test handling of equivalent performances). We run experiments for $M = 30$ questions, where each LLM "answers" all the questions in each trial according to its success probabilities $\pi_{\alpha 1}$. Panel (a) of Fig. 2 shows results without bootstrapping: we generate 1000 independent R matrices, each with 80 trials; for each step in the number of trials (from $N = 1$ to 80), we compute scores using Pass@k ($k = 2$, $k = 4$, and $k = 8$ with an unbiased estimator Eq. (21)), Bayes@N, a naive Pass^k variant $(1-(1-\hat{p})^k$, Eq. (22)), G-Pass@$k_{\tilde{\tau}}$ (Eq. (23) with $\tilde{\tau} = 0.5$), and mG-Pass@k(Eq. (24)), then derive rankings and compare them to the gold standard using Kendall's τ as a measure of rank correlation (where $\tau = 1$ indicates perfect alignment with the gold standard), and report the average τ over the 1000 R matrices. Note that we do not explicitly show average accuracy avg@N because it is equivalent to Bayes@N, as discussed in section 2.6. In practice, we are computationally limited to a small number of trials per question. To examine what happens with only $N = 80$ trials, we apply two methods of bootstrapping with replacement to the R matrix, allowing us to estimate how results differ from the ideal case with a large number of independent R matrices (panel a). For both methods, we generate 10,000 bootstrap replicates for each of the $N = 1$ to 80 trials, derived from a single R matrix. Panels (b) and (c) of Fig. 2 illustrate this using two bootstrapping schemes. In the first scheme (panel b, column-wise bootstrapping), we resample trial indices; in the second (panel c, row-wise bootstrapping), we resample answers independently for each question. In both cases, the resulting bootstrap replicates are used to recompute evaluation scores, rankings, and τ values, which are then averaged to produce smoothed convergence curves. The two bootstrapping approaches yield nearly identical behavior, and both closely match the baseline in panel (a). This demonstrates that the τ convergence behavior is robust and not sensitive to the ordering of answers in either the rows or columns of R. Though in our LLM mimic simulations, we do not have to use bootstrapping (since we can easily generate an arbitrarily large number of R matrices), in actual LLM experiments, we have limited trial data, and these results show that bootstrapping provides a viable way of estimating statistical properties like convergence.

As seen in Fig. 2, Bayes@N begins with relatively high agreement with the gold standard and converges much faster to $\tau = 1$ than Pass@k and its variants, which suffer from greater variance and bias at small N. All methods eventually converge to the same ranking, but their rates of convergence differ substantially. This makes the convergence rate a crucial factor when choosing between different LLM evaluation methods.

2.7.1 Ranking with Uncertainty

In section 2.5, we described how uncertainty estimates from the Bayesian approach can be used to evaluate the relative performance of two models. Here, we extend these ideas to incorporate uncertainty into the ranking of multiple models. We do this via our biased-coin LLM mimics, which we denote LLM$_\beta$ for $\beta = 1, \ldots, 11$, described in the previous section. To incorporate a chosen confidence interval in the ranking, we order their μ values from highest to lowest, choose the appropriate z threshold (for example $z = 1.645$ for 95% confidence in the ranking), and assign two consecutive methods the same ranking if the absolute z-score falls below this threshold.

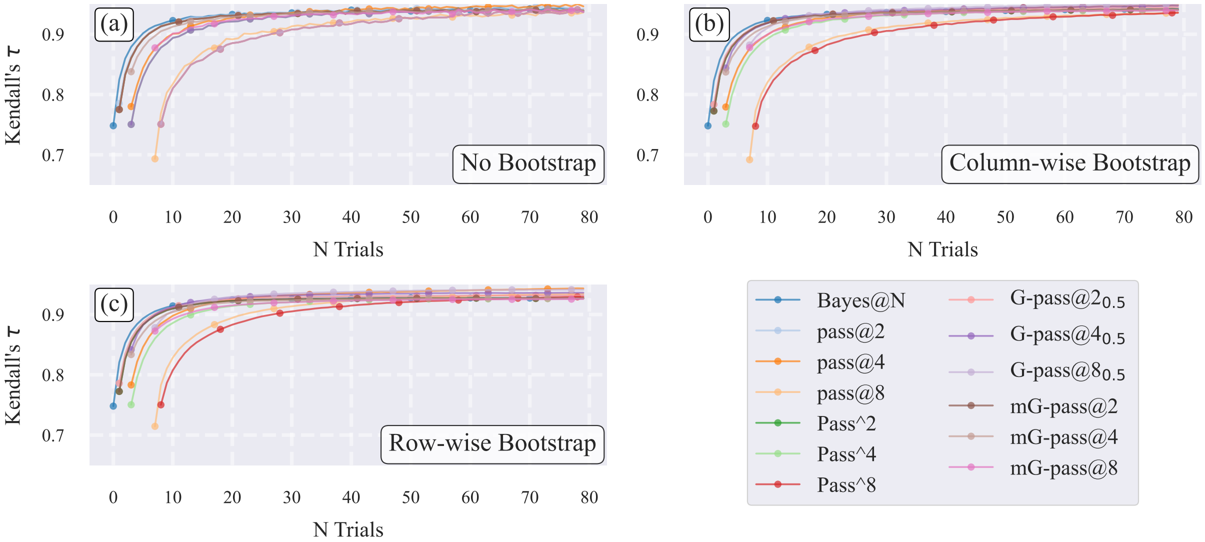

Figure 2: Kendall's τ rank correlation for various evaluation methods compared to the true ranking of 11 sets of biased coins (LLM mimics) with known mean success probabilities $\bar{\pi} = 0.2332$, 0.2545, 0.3604, 0.3642, 0.3642, 0.4466, 0.5418, 0.5276, 0.608 , 0.6213, 0.7327. The simulation evaluates methods including Pass@k ($k = 2, 4, 8$), Bayes@N, naive Passˆk, G-Pass@$k_{\tilde{\tau}}$ ($\tilde{\tau} = 0.5$), and mG-Pass@k across 1 to 80 trials. Panel a) shows τ results without bootstrapping, while panels b) and c) use two different bootstrapping approaches with 10^4 samples.

The first row of Table 1 shows the underlying gold standard ranking for all the LLM mimics, since in this case we know the true $\bar{\pi}$ values. Note the tie between LLM$_4$ and LLM$_5$, because their $\bar{\pi} = 0.3642$ is the same. The second row shows the Bayes@80 ranking without a confidence interval (CI) and the third row shows Bayes@80 incorporating the 95% CI. The Bayes@80 ranking without CI aligns with the gold standard, except for two differences: the order of LLM$_{10}$ and LLM$_9$ is swapped, and the tie between LLM$_5$ and LLM$_4$ is not captured, which is expected since this ranking relies solely on μ estimates without accounting for uncertainty σ. In contrast, the third row, which incorporates the CI, reveals multiple ties across several models. Interestingly, LLM$_{10}$ and LLM$_9$ are now indistinguishable at the 95% CI. Despite the fact that $N = 80$ would be an atypically large number of trials for an actual LLM evaluation, it is insufficient to confidently distinguish the small performance difference ($\bar{\pi} = 0.608$ vs. 0.6213) between the two models.

Table 1: Comparison of biased-coin LLM mimic rankings based on the gold standard, Bayes@80 without confidence interval (CI), and Bayes@80 with CI.

LLM mimic	LLM$_{11}$	LLM$_{10}$	LLM$_9$	LLM$_8$	LLM$_7$	LLM$_6$	LLM$_5$	LLM$_4$	LLM$_3$	LLM$_2$	LLM$_1$
Gold Standard	1	2	3	5	4	6	7	7	8	9	10
Bayes@80 (w/o CI)	1	3	2	5	4	6	7	8	9	10	11
Bayes@80 (w/ CI)	1	2	2	3	3	4	5	5	5	6	7

To quantify the trials needed to reliably separate models with closely matched performance, we simulated the probability of correctly ranking LLM$_{10}$ above LLM$_9$ as a function of the number of trials N, shown in the left panel of Fig. 3. At $N = 80$, the probability of obtaining the correct ranking is 83.7%. The right panel plots the absolute z-score versus N; at $N = 80$, the $z \sim 1.14$, corresponding to approximately 87% confidence (though the plots exhibit some noise due to simulation variability). These values closely align with the empirical probabilities in the left panels.

We also determined the minimum sample size N needed to achieve z-scores of 1.645 and 1.96, corresponding to CI of approximately 95% and 97.5%, respectively, for distinguishing between models. These thresholds occur at about $N = 199$ and $N = 285$. At these values, the simulated probability of correctly ranking the models is 94.7% and 96.9%, respectively, which is closely consistent with expectations given the inherent noise in the results. These results underscore the computational cost of distinguishing models whose true performance metrics differ only slightly. In our biased-coin setup, the underlying success probabilities were $\bar{\pi}_9 = 0.608$ and $\bar{\pi}_{10} = 0.6213$, yet reliably establishing this distinction requires nearly 200 trials. Such large sample requirements highlight the importance of considering both uncertainty and convergence rates when interpreting ranking-based evaluations.

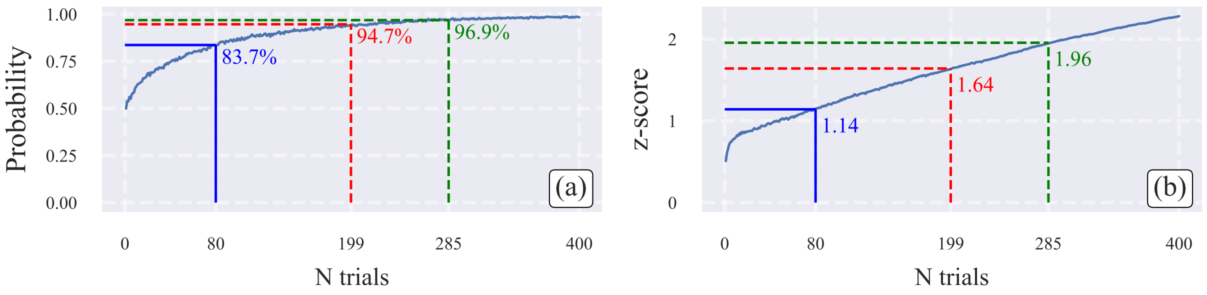

Figure 3: (a) Probability of correctly ranking LLM_{10} above LLM_9 using Bayes@N in the biased-coin simulations, shown as a function of trial count N. The probability is 83.7% at $N = 80$, increases to $\sim 94.7\%$ at $N = 199$, and reaches 96.9% at $N = 285$. (b) Corresponding absolute z-scores as a function of N, with values of ~ 1.14 at $N = 80$, 1.645 at $N = 199$ (95% confidence), and 1.96 at $N = 285$ (97.5% confidence).

3 Experiments

In this section, we empirically validate our proposed evaluation methods using real-world datasets, focusing on ranking LLMs for mathematical reasoning tasks. We employ bootstrapping to compute the expected value of each evaluation score at a given N. First, we present rankings of LLMs on the AIME'24, AIME'25, BrUMO'25, and HMMT'25 datasets without accounting for variance, based solely on evaluation scores (with ties occurring when scores are identical). Subsequently, we demonstrate how incorporating uncertainty in these scores can alter rankings across different datasets. Building on the discussion in section 2.7, we adopt the ranking derived from avg@80 (Pass@1) or Bayes@80 (uniform prior Bayesian estimator) at $N = 80$ (the total number of trials conducted per dataset) as our gold standard for comparing current LLMs, noting their equivalence in rankings (as proven in Section 2.6). For each N from 1 to 80 (with Pass@k and similar methods starting from $N = k$ to avoid computation with insufficient samples), we compare the rankings produced by various evaluation methods against this gold standard, reporting the average Kendall's τ over 10^4 bootstrapped resamples to estimate the expected rank correlation at each step (assuming independence among questions and trials).

3.1 Convergence to Gold Standard

To assess the ability of different evaluation methods to compare the performance of different LLMs, we plot the average Kendall's τ against the gold standard as a function of the number of trials N in Fig. 4, combining results from AIME 2025 (panel a), AIME 2024 (panel b), HMMT'25(panel c), and BrUMO'25(panel d). Across all datasets, the Bayes@N and avg@N curves overlap completely (so we only plot Bayes@N) and demonstrate the fastest convergence to high τ values, indicating robust alignment with the gold standard even in low-sample regimes. For AIME'24, HMMT'25, and BrUMO'25, Bayes@N achieves $\tau > 0.95$ by $N = 10$ and approaches $\tau \sim 1$ at $N \approx 80$, but for AIME'25, it achieves $\tau > 0.90$ by $N = 10$ and converges to $\tau \sim 0.95$ at $N = 80$.

In contrast, Pass@k variants ($k = 2, 4, 8$) and their variations (e.g., naive Passˆk, G-Pass@$k_{\tilde{\tau}}$ with $\tilde{\tau} = 0.5$, mG-Pass@k) start with lower Kendall's τ compared to Bayes@N and converge more slowly in all datasets except for AIME'25. In AIME'24, HMMT'25, and BrUMO'25, Bayes@N converges faster to the gold standard, and all other evaluation methods converge to lower τ compared to Bayes@N. This confirms our expectations from the simulated biased coins in Section 2.7. However, in AIME 2025, we observe that G-pass@$2_{\tilde{\tau}=0.5}$ and G-pass@$4_{\tilde{\tau}=0.5}$ converge faster than Bayes@N but notably to a lower Kendall's τ. On the other hand, Bayes@N converges faster than the rest of the methods and also converges to the highest τ compared to all methods. These results mirror our biased-coin simulations, confirming that the Bayesian approach satisfies our gold standard criteria—low uncertainty, minimal ties, and rapid convergence—across diverse math reasoning benchmarks.

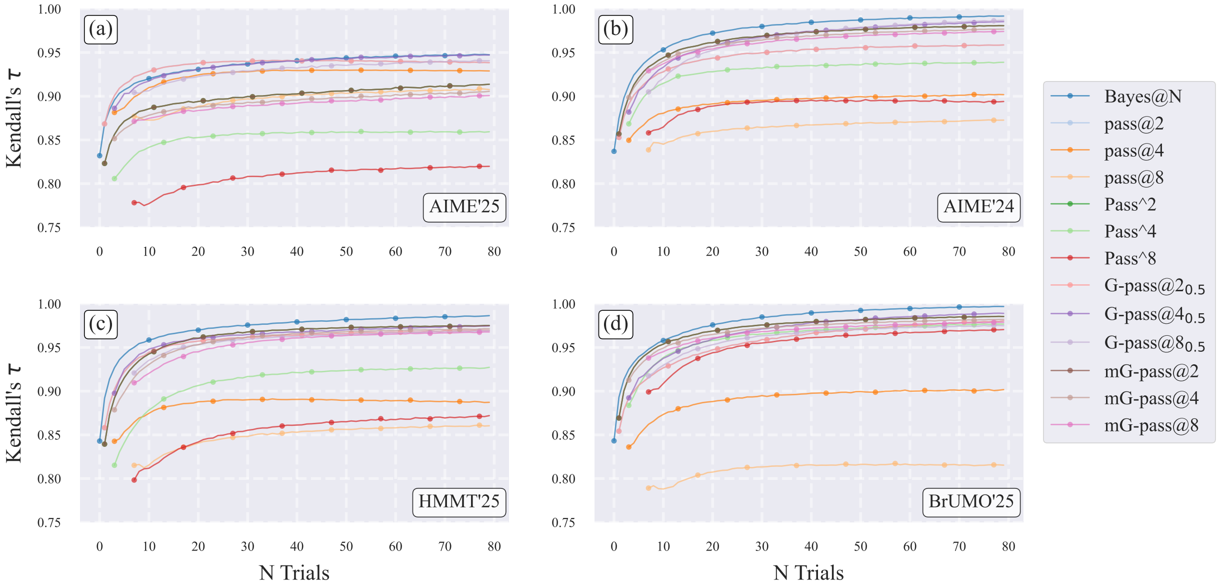

Figure 4: Average Kendall's τ correlation between rankings produced by various evaluation methods and the gold standard (derived from Bayes@80, or equivalently avg@80), as a function of the number of trials N. Results are averaged over 10^4 bootstrapped resamples for each dataset: (a) AIME'25, (b) AIME'24, (c) HMMT'25, and (d) BrUMO'25. Methods include Bayesian estimation Bayes@N , Pass@k ($k = 2, 4, 8$), naive Pass^k, G-Pass@$k_{\tilde{\tau}}$ ($\tilde{\tau} = 0.5$), and mG-Pass@k.

3.2 Rankings With Confidence Intervals

Following the methodology of section 2.7.1, we compare model rankings across four datasets (AIME'25, AIME'24, HMMT'25, and BrUMO'25) using Bayes@80 as the gold standard (see Fig. 4). Table 2 summarizes these comparisons by reporting, for each dataset, two versions of the ranking: the rank *with* a 95% confidence interval (CI) and the rank *without* CI. The "w/ CI" rank accounts for uncertainty in the Bayes@80 scores and therefore allows models with overlapping CIs to share the same rank; the "w/o CI" rank is the strict ordering determined by the point estimates of Bayes@80 for that dataset.

Table 2 shows that the point-estimate rankings (w/o CI) for AIME'24 and AIME'25 are very similar, and both share partial similarities with HMMT'25; by contrast, BrUMO'25 exhibits a markedly different ordering. Across all four datasets, the model Qwen3-30B-A3B-Thinking-2507 consistently attains the top position, and this superiority is statistically distinguishable at the 95% CI level in every dataset. The relative order of the remaining models, however, varies between datasets.

Considering the rankings that incorporate 95% CIs, we observe different degrees of ambiguity across datasets: AIME'25 exhibits three ties, AIME'24 and HMMT'25 each show two ties, and BrUMO'25 shows only one tie. This pattern indicates that the Bayes@80 gold standard for BrUMO'25 has the lowest uncertainty (fewest ties) under our trial budget, while AIME'25 shows the highest uncertainty. Intuitively, higher uncertainty in the gold-standard scores implies that more trials would be required to empirically produce a more stable ranking; lower uncertainty means we can be more confident in the estimated gold standard given the current number of trials.

These differences in uncertainty explain the convergence behavior observed in Fig. 4: BrUMO'25 quickly reaches $\tau = 1$ by $N = 80$, HMMT'25 and AIME'24 converge to values close to 1 (but slightly lower than BrUMO'25), and AIME'25 only attains $\tau \approx 0.95$ at $N = 80$. We also note that G-Pass@2 tends to converge faster but to a lower τ; this may be attributable to higher variance in its score estimates across trials, making its relative ordering more sensitive to gold-standard uncertainty. For example, Table 2 shows that gpt-oss-20b-high and gpt-oss-20b-medium are tied in all datasets except BrUMO'25, where they are distinguishable at the 95% CI level. Examining the z-scores between these two models yields values of 0.0427, 0.9616, 0.6374, and 1.8272 for AIME'25, AIME'24, HMMT'25, and BrUMO'25, respectively. These correspond to probabilities of correctly ranking gpt-oss-20b-medium and gpt-oss-20b-high from one another, of approximately 51.7%, 83.2%, 73.8%, and 96.6% with $N = 80$ trials.

Table 2: Rankings for four datasets. Models are listed in the order of their gold-standard ranking (Bayes@80 point estimates, i.e., without uncertainty) for AIME'25. Each dataset column gives the rank with a 95% confidence interval (left) and the rank without CI (right).

Model	AIME'25		AIME'24		HMMT'25		BrUMO'25	
	w/ CI	w/o CI	w/ CI	w/o CI	w/ CI	w/o CI	w/ CI	w/o CI
Qwen3-30B-A3B-Thinking-2507	1	1	1	1	1	1	1	1
gpt-oss-20b-high	2	2	2	3	2	3	4	4
gpt-oss-20b-medium	2	3	2	2	2	2	5	5
gpt-oss-20b-low	3	4	4	5	7	9	8	9
OpenThinker2-32B	3	5	3	4	3	4	2	2
LIMO-v2	4	6	5	6	4	5	3	3
EXAONE-4.0-1.2B	5	7	6	7	4	6	7	7
OpenThinker3-1.5B	6	8	7	8	5	7	6	6
OpenReasoning-Nemotron-1.5B	6	9	8	9	6	8	7	8
DeepSeek-R1-Distill-Qwen-1.5B	7	10	9	11	8	10	10	11
Sky-T1-32B-Flash	8	11	9	10	9	11	9	10

3.3 Convergence

In this section, we investigate the convergence of model rankings in more detail, building on the showcase figure presented in the introduction (Fig. 1). We define convergence@n as the smallest number of trials N where the ranking of LLM models, derived from the observed outcomes up to N, matches the *gold standard* ranking from Bayes@80 on the actual 80-trial matrix (without bootstrapping), and remains unchanged with additional trials. Lower convergence@n values indicate faster convergence, implying that fewer trials are sufficient to achieve stable rankings. As detailed in the figure's caption, it displays the probability mass functions (PMFs) of convergence@n for each method across the datasets. These PMFs are empirically estimated by generating 10^5 row-wise bootstrap replicates through resampling the $N_{\max}$ trials, then for each replicate, cumulatively evaluating the ranking at every N (from 1 to 80) and identifying the minimal n where the ranking stabilizes to the *gold standard*. This process captures the distribution of convergence points under repeated sampling, reflecting the inherent uncertainty in finite-sample rankings due to stochastic trial outcomes.

This bootstrapping approach provides a distribution over possible convergence points (n), offering insights into the variability and reliability of each evaluation method: traditional Pass@k (for $k = 2, 4, 8$) versus our Bayes@N. A lower mean convergence@n signifies more cost-effective convergence, while failure to converge within 80 trials (as seen in AIME'25) indicates more trials are needed to confidently rank LLMs or we must include CI for a reliable ranking.

The key takeaways from Fig. 1, as summarized in its caption, underscore the superiority of the Bayes@N: it converges reliably on all datasets except AIME'25, often with fewer trials than Pass@k. For instance, on HMMT'25 and BrUMO'25, Bayes@N achieves mean convergence at approximately 44.2 and 27.1 trials, respectively, compared to around 69.5 and 48.5 for the Pass@k family. The right panel of the figure further illustrates this through an example ranking from a bootstrap replicate, emphasizing differences in convergence for AIME'25 and BrUMO'25. See Section C (Fig. 6) for the corresponding cumulative distribution functions.

To complement the aggregate view from Fig. 1, we analyze the *worst-case* bootstrap replicates, those requiring the maximum trials to stabilize rankings. Fig. 5 presents these trajectories as competition-style rankings, with each line tracing a model's rank position across incrementally added trials. Convergence is achieved when the ranking order remains unchanged for all subsequent trials.

For AIME'24 (panel a), the ranking converges at trial 75, reflecting efficient convergence under Bayes@N. For AIME'25 (panel b), no convergence is observed within 80 trials, underscoring persistent instability and the need for more trials and confidence intervals. For BrUMO'25 (panel c), convergence occurs at trial 68, demonstrating robust convergence. For HMMT'25 (panel d), the ordering converges at trial 78, indicating a slightly higher trial requirement. These worst-case examples illustrate Bayes@N's tendency to achieve stable rankings with fewer trials than typical Pass@k evaluations, making it a more cost-effective option for reliable model comparison. When a ranking does not

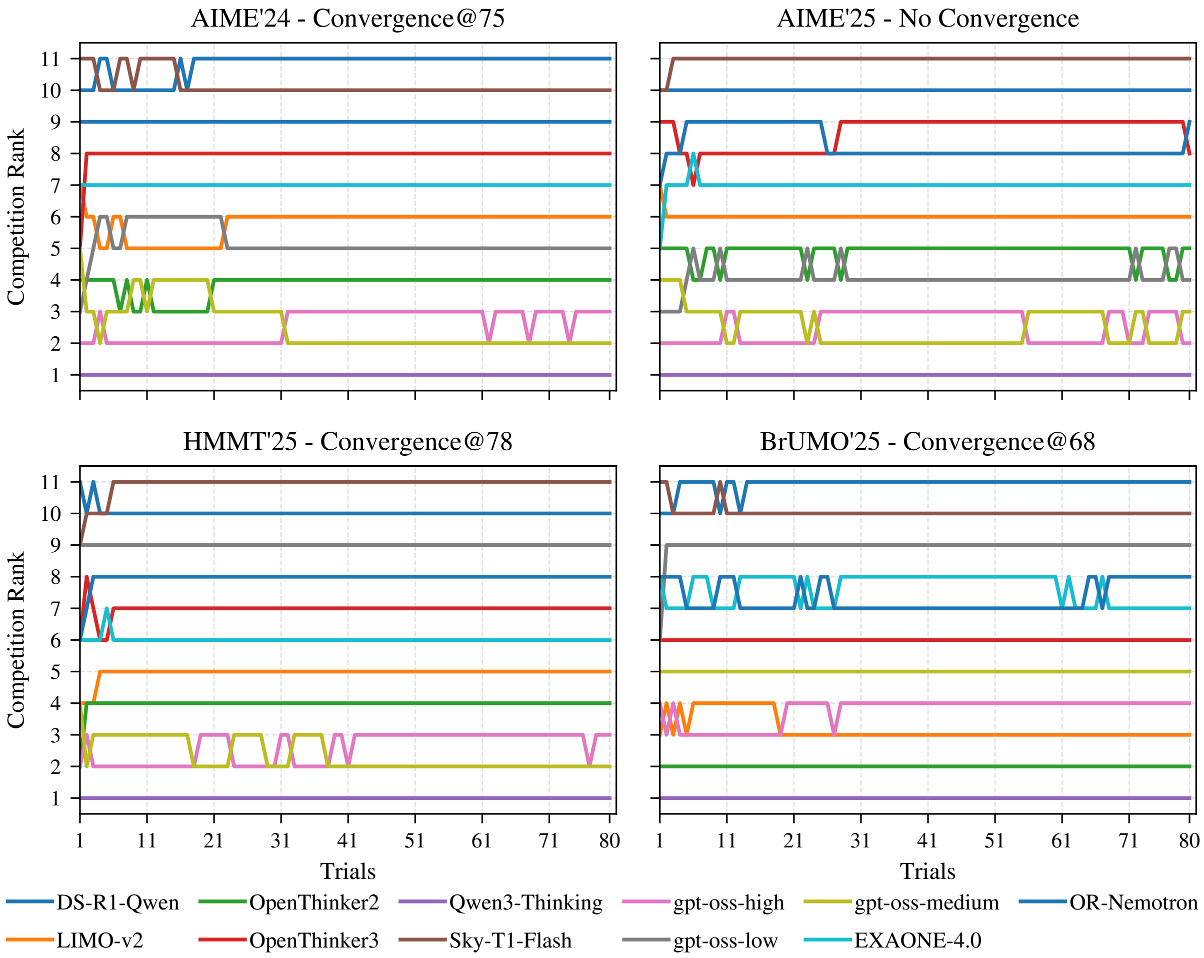

Figure 5: Worst-case bootstrap rank trajectories for four models. Each line shows a ranking as trials are added; horizontal segments indicate the ranking at $N = 80$. Convergence is defined as the minimal N after which the ranking remains unchanged. (a) AIME'24: converges at $N = 75$. (b) AIME'25: no convergence observed within 80 trials. (c) BrUMO'25: converges at $N = 68$. (d) HMMT'25: converges at $N = 78$.

converge within the trial budget (as for AIME'25), we can incorporate confidence intervals to quantify uncertainty and to estimate the minimum N required for a reliable ranking (see section 2.7.1).

3.4 Rubric-Aware Categorical Evaluation

While evaluation is often reduced to binary correctness, this simplification discards valuable signals that capture other aspects of model behavior. For instance, LLM outputs can be assessed not only on correctness but also on whether they are well-structured, coherent, or exhibit step-by-step reasoning in mathematical tasks. In practice, evaluators routinely record richer dimensions such as format compliance, calibration of confidence, degenerate outputs, out-of-distribution (OOD) behavior, and verifier scores. This limitation is especially important for reasoning models, where overthinking (45) inflates token usage without corresponding gains in reliability. Bayes@N provides a principled way to capture these richer outcomes. By treating per-item results as categorical rather than binary, the approach aligns more closely with actual goals while preserving statistical rigor and transparency. This method enables a nuanced understanding of model performance across diverse dimensions, offering insights into trade-offs between correctness, efficiency, and robustness. For a comprehensive discussion of the categorical Bayesian evaluation framework, including base signals, schema definitions, and their impact on model rankings, see Appendix D.

Table 3: Comparison of the Bayesian framework and other evaluation methods.

Evaluation method (N trials)	**Convergence**	**Confidence interval**	**Prior knowledge**	**Categorical**
Pass@k and alternatives	✗	✗	✗	✗
avg@N	✓	Limited (via bootstrap/binomial CIs)	✗	✗
Bayes@N	✓ (Figs. 1 and 5)	✓ (Fig. 3, Table 1,2)	✓	✓ (Sec.3.4)

4 Related Work

Functional-correctness evaluation with Pass@k became standard in code generation with HumanEval (OpenAI Codex): generate k samples, a task is solved if any sample passes unit tests, and estimate the overall rate with an unbiased estimator that requires producing $n > k$ samples per task (40). Although Pass@k was initially introduced in the context of coding, it later became the de facto choice to evaluate LLMs not only on math reasoning tasks (46; 47; 48; 49; 50; 51; 52; 53; 54; 55) but also on safety evaluations spanning agent red-teaming, jailbreaks, and backdoor analyses (56; 57; 58; 59; 60; 61). For a broader review of these metrics and their variants, see Appendix E. Beyond standard Pass@k, *pass^k* quantifies reliability across k i.i.d. trials for agents, while the generalized *G-pass@*k_τ continuum (and its area-under-τ summary *mG-Pass*) jointly assess potential and stability in reasoning outputs (62; 54; 63).

Efforts like HELM advance holistic, transparent evaluation across scenarios and metrics (5), while practice guidelines distill reproducibility pitfalls and prescribe multi-run, uncertainty-aware reporting with fixed prompts, decoding, and dataset/version control (64). The LM Evaluation Harness offers standardized, reproducible frameworks to implement these recommendations (64). It addresses the need for calibrated uncertainty in small-sample reasoning by employing exact methods for error quantification in binomial settings.

The last category of related work focuses on measuring uncertainty in LLM evaluation. These works converge on interval-aware, small-sample-valid reporting rather than CLT/Wald error bars. Bowyer et al. show that CLT-based intervals *miscalibrate* on small benchmarks and advocate small-n-appropriate frequentist or Bayesian intervals for reliable comparisons (44). A Bayesian alternative models capability as a latent success probability and reports posterior credible intervals that remain informative with limited trials, yielding more stable rankings (33). In judge-based settings, *Judging LLMs on a Simplex* places model and judge behavior on the probability simplex, enabling uncertainty-aware comparisons and highlighting how distributional structure matters for evaluation (65). Beyond bespoke LLM metrics, prediction-powered inference supplies general procedures for valid confidence intervals that leverage model predictions to reduce labeled-sample requirements (66). Finally, in adjacent retrieval evaluation with LLM-generated assessments, Oosterhuis et al. construct reliable confidence intervals and demonstrate that calibrated uncertainty, rather than point estimates, should guide decisions, reinforcing this shift for LLM evaluation more broadly (67).

5 Conclusion: Strengths, Limitations & Future Directions

The overall benefits of the Bayesian framework are summarized in Table 3: it provides fast convergence, analytical estimates of confidence intervals, and incorporation of prior knowledge and categorical results. However it is worth noting that our approach quantifies *statistical* uncertainty from finite samples; it does not fix dataset bias, distribution shift, or rubric misspecification. Results therefore depend on the chosen benchmark, prompts, and inference settings (hardware). Although we have validated our approach with biased-coin LLM mimic simulations, together with experiments using actual LLMs (up to $N_{\text{max}} = 80$ trials across four tasks and 11 models), more extensive evaluations may be constrained by computing and academic budgets.

The focus of the current work was the simplest version of the Bayesian approach, using a uniform prior, which provides a conservative and reproducible starting point. But the theory allows for more complex, informative priors, and this opens up a rich vein of future directions that should be systematically explored: for example priors from past runs, domain- or task-conditioned priors, and expert-elicited priors. These have the potential of accelerating convergence even further, but must be chosen and reported carefully. Clear guidance and tools for prior elicitation will hopefully ensure that gains in sample efficiency do not come at the cost of hidden bias.

Ethics Statement

This research relies only on publicly available, non-personal benchmarks; no human subjects, user data, or PII are involved. Potential misuse includes cherry-picking priors, rubrics, or samples to exaggerate performance. To prevent this, use of Bayes@N with user-defined priors requires clear documentation and reporting of posterior credible intervals.

Reproducibility statement

To ensure reproducibility, detailed implementation instructions are provided in Appendix F.

Acknowledgment

This research was supported in part by NSF awards 2117439, 2112606, and 2320952.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.03762.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[3] StackOverflow. Stack Overflow Developer Survey 2025: AI and Developer Tools, 2025. URL https://survey.stackoverflow.co/2025/ai. Accessed: 2025-09-24.

[4] Nestor Maslej, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, Anka Reuel, Erik Brynjolfsson, John Etchemendy, et al. Artificial intelligence index report 2025. *arXiv preprint arXiv:2504.07139*, 2025.

[5] Percy Liang, Rishi Bommasani, et al. Holistic evaluation of language models. *arXiv:2211.09110*, 2022. URL https://arxiv.org/abs/2211.09110.

[6] Dan Hendrycks, Collin Burns, Steven Basart, et al. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2009.03300.

[7] Aarohi Srivastava, Abhinav Rastogi, et al. Beyond the Imitation Game: Quantifying and Extrapolating the Capabilities of Language Models (BIG-bench). *arXiv:2206.04615*, 2022. URL https://arxiv.org/abs/2206.04615.

[8] Jared Kaplan, Sam McCandlish, Tom Henighan, et al. Scaling laws for neural language models. *arXiv:2001.08361*, 2020. URL https://arxiv.org/abs/2001.08361.

[9] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, et al. Training compute-optimal large language models. *arXiv:2203.15556*, 2022. URL https://arxiv.org/abs/2203.15556.

[10] Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/pdf?id=_VjQlMeSB_J.

[11] Long Ouyang, Jeff Wu, Xu Jiang, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

[12] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale, 2022. URL https://arxiv.org/abs/2208.07339.

[13] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers, 2022. URL https://arxiv.org/abs/2210.17323.

[14] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural networks. In *NeurIPS*, 2015. URL https://papers.nips.cc/paper/5784-learning-both-weights-and-connections-for-efficient-neural-network.

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

[16] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *SOSP*, 2023. URL https://arxiv.org/abs/2309.06180.

[17] Tianyi Zhang, Jonah Yi, Zhaozhuo Xu, and Anshumali Shrivastava. Kv cache is 1 bit per channel: Efficient large language model inference with coupled quantization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 3304–3331. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/05d6b5b6901fb57d2c287e1d3ce6d63c-Paper-Conference.pdf.

[18] Hailin Zhang, Xiaodong Ji, Yilin Chen, Fangcheng Fu, Xupeng Miao, Xiaonan Nie, Weipeng Chen, and Bin Cui. Pqcache: Product quantization-based kvcache for long context llm inference. *Proc. ACM Manag. Data*, 3 (3), June 2025. doi: 10.1145/3725338. URL https://doi.org/10.1145/3725338.

[19] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*, 2021. URL https://arxiv.org/abs/2106.09685.

[20] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *NeurIPS*, 2017. URL https://arxiv.org/abs/1706.03741.

[21] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

[22] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[23] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, et al. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

[24] American Invitational Mathematics Examination (AIME) — official description. Mathematical Association of America, 2025. URL https://maa.org/maa-invitational-competitions/. 15 questions, 3 hours.

[25] Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility. *arXiv preprint arXiv:2504.07086*, 2025.

[26] Junnan Liu, Hongwei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao, Wenwei Zhang, Songyang Zhang, and Kai Chen. Are your llms capable of stable reasoning? *arXiv preprint arXiv:2412.13147*, 2024.

[27] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *ICLR*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH. arXiv:1904.09751 (2019).

[28] Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The Hitchhiker's Guide to Testing Statistical Significance in NLP. In *ACL*, pages 1383–1392, 2018. URL https://aclanthology.org/P18-1128/.

[29] Alexander Yeh. More accurate tests for the statistical significance of result differences. In *COLING*, 2000. URL https://aclanthology.org/C00-2137/.

[30] Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. Show your work: Improved reporting of experimental results. In *EMNLP-IJCNLP*, 2019. URL https://aclanthology.org/D19-1224/.

[31] Lianmin Zheng, Wei-Lin Chiang, Yingbo Sheng, and et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685*, 2023. URL `https://arxiv.org/abs/2306.05685`.

[32] Guande Chen, Kai Shen, Saurav Shah, and et al. Humans or LLMs as the Judge? A Study on Judgement Bias. In *EMNLP*, 2024. URL `https://aclanthology.org/2024.emnlp-main.474.pdf`.

[33] Xiao Xiao, Yu Su, Sijing Zhang, Zhang Chen, Yadong Chen, and Tian Liu. Confidence in large language model evaluation: A bayesian approach to limited-sample challenges. *arXiv preprint arXiv:2504.21303*, 2025.

[34] Dustin Hayden and Thomas Armitage. Straightforward bayesian a/b testing with dirichlet posteriors. *arXiv preprint arXiv:2508.08077*, 2025.

[35] Mathematical Association of America. American invitational mathematics examination (aime). `https://maa.org/maa-invitational-competitions/`, 2024. Official MAA page for the AIME competition (covers AIME 2024).

[36] Mathematical Association of America. American invitational mathematics examination (aime). `https://maa.org/maa-invitational-competitions/`, 2025. Official MAA page for the AIME competition (covers AIME 2025).

[37] Harvard–MIT Mathematics Tournament. Hmmt february 2025 archive (problems and solutions). `https://www.hmmt.org/www/archive/282`, 2025. Official HMMT archive page for February 2025 competition.

[38] Brown University Math Olympiad Organizers. Brown university math olympiad (brumo). `https://www.brumo.org/tournament-info`, 2025. Official BrUMO website with tournament information (Apr 4–5, 2025).

[39] Uri Dalal, Meirav Segal, Zvika Ben-Haim, Dan Lahav, and Omer Nevo. Leveraging LLM Inconsistency to Boost Pass@ k Performance. *arXiv preprint arXiv:2505.12938*, 2025.

[40] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[41] Brendan Leigh Ross, NoÃĢl Vouitsis, Atiyeh Ashari Ghomi, Rasa Hosseinzadeh, Ji Xin, Zhaoyan Liu, Yi Sui, Shiyi Hou, Kin Kwan Leung, Gabriel Loaiza-Ganem, et al. Textual Bayes: Quantifying Uncertainty in LLM-Based Systems. *arXiv preprint arXiv:2506.10060*, 2025.

[42] Roman Vashurin, Maiya Goloburda, Albina Ilina, Aleksandr Rubashevskii, Preslav Nakov, Artem Shelmanov, and Maxim Panov. Uncertainty Quantification for LLMs through Minimum Bayes Risk: Bridging Confidence and Consistency. *arXiv preprint arXiv:2502.04964*, 2025.

[43] Edwin T Jaynes. *Probability theory: The logic of science*. Cambridge university press, 2003.

[44] Sam Bowyer, Laurence Aitchison, and Desi R Ivanova. Position: Don't Use the CLT in LLM Evals With Fewer Than a Few Hundred Datapoints. *arXiv preprint arXiv:2503.01747*, 2025.

[45] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

[46] 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.

[47] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[48] Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 7821–7846. Curran Associates, Inc., 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/0ef1afa0daa888d695dcd5e9513bafa3-Paper-Conference.pdf`.

[49] Bingbin Liu et al. TinyGSM: achieving >80% on GSM8K with small language models. 2023.

[50] Hyeongdon Hwang et al. Self-explore: Enhancing mathematical reasoning in large language models by finding the first pit. In *Findings of EMNLP*, 2024. URL `https://aclanthology.org/2024.findings-emnlp.78/`.

[51] Yan Yang et al. Weak-to-strong reasoning. In *Findings of EMNLP*, 2024. URL `https://aclanthology.org/2024.findings-emnlp.490/`.

[52] Niklas Muennighoff et al. s1: Simple test-time scaling. 2025.

[53] Fangchen Chen et al. Rethinking fine-tuning when scaling test-time compute. 2025.

[54] Junnan Liu et al. Are Your LLMs Capable of Stable Reasoning? 2024.

[55] LG Research. EXAONE Deep: Reasoning Enhanced Language Models. 2025.

[56] Eyal Nakash et al. Effective red-teaming of policy-adherent agents. 2025.

[57] Hojjat Aghakhani et al. Trojanpuzzle: Covertly poisoning code-suggestion models. In *IEEE Symposium on Security and Privacy*, 2024.

[58] Hongyi Liu, Shaochen Zhong, Xintong Sun, Minghao Tian, Mohsen Hariri, Zirui Liu, Ruixiang Tang, Zhimeng Jiang, Jiayi Yuan, Yu-Neng Chuang, et al. LoRATK: LoRA Once, Backdoor Everywhere in the Share-and-Play Ecosystem. *arXiv preprint arXiv:2403.00108*, 2024.

[59] Zheng Yan et al. An LLM-Assisted Easy-to-Trigger Backdoor Attack on Code LMs. In *USENIX Security Symposium*, 2024.

[60] Jiaheng Wang et al. RTL-Breaker: Backdoor Attacks on HDL Code Generation. 2024.

[61] Rylan Schaeffer et al. How do large language monkeys get their power (laws)? In *Proceedings of ICML*, 2025. Oral.

[62] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. 2024. doi: 10.48550/arXiv.2406.12045. Introduces the pass^k metric.

[63] Junnan Liu, Hongwei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao, Wenwei Zhang, Songyang Zhang, and Kai Chen. Are your llms capable of stable reasoning? In *Findings of ACL*, 2025. URL `https://aclanthology.org/2025.findings-acl.905/`. Camera-ready version detailing G-Pass@k$_\tau$ and mG-Pass.

[64] Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024. URL `https://arxiv.org/abs/2405.14782`.

[65] Patrick Vossler, Fan Xia, Yifan Mai, and Jean Feng. Judging llms on a simplex. *arXiv preprint arXiv:2505.21972*, 2025.

[66] Anastasios N Angelopoulos, Stephen Bates, Clara Fannjiang, Michael I Jordan, and Tijana Zrnic. Prediction-powered inference. *Science*, 382(6671):669–674, 2023.

[67] Harrie Oosterhuis, Rolf Jagerman, Zhen Qin, Xuanhui Wang, and Michael Bendersky. Reliable confidence intervals for information retrieval evaluation using generative ai. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2307–2317, 2024.

[68] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

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

[70] 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.

[71] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL `https://aclanthology.org/P02-1040/`.

[72] Shuo Ren, Daya Guo, Shuai Lu, Long Zhou, Shujie Liu, Duyu Tang, Neel Sundaresan, Ming Zhou, Ambrosio Blanco, and Shuai Ma. Codebleu: a method for automatic evaluation of code synthesis. *arXiv preprint arXiv:2009.10297*, 2020.

[73] Sumith Kulal, Panupong Pasupat, Kartik Chandra, Mina Lee, Oded Padon, Alex Aiken, and Percy S Liang. Spoc: Search-based pseudocode to code. *Advances in Neural Information Processing Systems*, 32, 2019.

[74] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[75] 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.

[76] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[77] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

[78] NovaSky Team. Think less, achieve more: Cut reasoning costs by 50 https://novasky-ai.github.io/posts/reduce-overthinking, 2025. Accessed: 2025-01-23.

[79] Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

[80] OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL `https://arxiv.org/abs/2508.10925`.

[81] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. LIMO: Less is More for Reasoning, 2025. URL `https://arxiv.org/abs/2502.03387`.

[82] LG AI Research. Exaone 4.0: Unified large language models integrating non-reasoning and reasoning modes. *arXiv preprint arXiv:2507.11407*, 2025.

[83] 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. URL `https://openreview.net/forum?id=8LhnmNmUDb`.

[84] Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. AIMO-2 Winning Solution: Building State-of-the-Art Mathematical Reasoning Models with OpenMathReasoning dataset, 2025. URL `https://arxiv.org/abs/2504.16891`.

[85] Wasi Uddin Ahmad, Somshubra Majumdar, Aleksander Ficek, Sean Narenthiran, Mehrzad Samadi, Jocelyn Huang, Siddhartha Jain, Vahid Noroozi, and Boris Ginsburg. OpenCodeReasoning-II: A Simple Test Time Scaling Approach via Self-Critique, 2025. URL `https://arxiv.org/abs/2507.09075`.

[86] Wasi Uddin Ahmad, Sean Narenthiran, Somshubra Majumdar, Aleksander Ficek, Siddhartha Jain, Jocelyn Huang, Vahid Noroozi, and Boris Ginsburg. OpenCodeReasoning: Advancing Data Distillation for Competitive Coding. 2025. URL `https://arxiv.org/abs/2504.01943`.

[87] Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. OpenThoughts: Data Recipes for Reasoning Models, 2025. URL https://arxiv.org/abs/2506.04178.

[88] Shudong Liu, Hongwei Liu, Junnan Liu, Linchen Xiao, Songyang Gao, Chengqi Lyu, Yuzhe Gu, Wenwei Zhang, Derek F. Wong, Songyang Zhang, and Kai Chen. CompassVerifier: A Unified and Robust Verifier for Large Language Models. 2025.

[89] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable., 2022. URL https://github.com/huggingface/accelerate.

[90] Tianyi Zhang, Mohsen Hariri, Shaochen Zhong, Vipin Chaudhary, Yang Sui, Xia Hu, and Anshumali Shrivastava. 70% size, 100% accuracy: Lossless llm compression for efficient gpu inference via dynamic-length float. arXiv preprint arXiv:2504.11651v2, 2025. URL https://arxiv.org/abs/2504.11651. Accepted in NeurIPS 2025.

CONTENTS

A Derivation of Bayesian Estimator and Uncertainty

As described in the main text, the Bayesian framework is built on two quantities. The first is $\mu(R)$, the average of $\bar{\pi}$ over the joint posterior for all the questions:

$$\mu(R) = \int_{\Delta} d\boldsymbol{\pi}_1 \cdots \int_{\Delta} d\boldsymbol{\pi}_M \, \bar{\pi} \prod_{\alpha=1}^{M} \mathcal{P}(\boldsymbol{\pi}_\alpha | \boldsymbol{R}_\alpha), \tag{2}$$

where the integration region Δ is the probability simplex defined as the set of all possible $(C+1)$-dimensional vectors $\boldsymbol{p}$ such that $\sum_{k=0}^{C} p_k = 1$. The second is the variance $\sigma^2(R)$ associated with our Bayesian estimator,

$$\sigma^2(R) = \int_{\Delta} d\boldsymbol{\pi}_1 \cdots \int_{\Delta} d\boldsymbol{\pi}_M \, (\bar{\pi} - \mu(R))^2 \prod_{\alpha=1}^{M} \mathcal{P}(\boldsymbol{\pi}_\alpha | \boldsymbol{R}_\alpha). \tag{3}$$

Our derivation of closed-form expressions for μ and σ builds on the generalized ($C > 1$) and original ($C = 1$) Laplace rule of succession theory from (43), recovering those results in the special case of a single question ($M = 1$). We start with Bayes' rule for each row of R:

$$\mathcal{P}(\boldsymbol{\pi}_\alpha | \boldsymbol{R}_\alpha) = \frac{\mathcal{P}(\boldsymbol{R}_\alpha | \boldsymbol{\pi}_\alpha)\mathcal{P}(\boldsymbol{\pi}_\alpha)}{\mathcal{P}(\boldsymbol{R}_\alpha)}. \tag{4}$$

The likelihood $\mathcal{P}(\boldsymbol{R}_\alpha | \boldsymbol{\pi}_\alpha)$ is a $(C+1)$-category multinomial distribution over N trials, with the probability distribution function:

$$\mathcal{P}(\boldsymbol{R}_\alpha | \pi_\alpha) = \frac{N!}{n_{\alpha 0}! n_{\alpha 1}! \cdots n_{\alpha C}!} \prod_{k=0}^{C} (\pi_{\alpha k})^{n_{\alpha k}}, \tag{5}$$

where $n_{\alpha k} = \sum_{i=1}^{N} \delta_{k, R_{\alpha i}}$, $\boldsymbol{n}_\alpha$ is the vector with elements $n_{\alpha k}$, and $\delta_{i,j}$ is the Kronecker delta.

The prior $\mathcal{P}(\boldsymbol{\pi}_\alpha)$ is chosen as the conjugate prior of the multinomial, a Dirichlet distribution $\mathcal{P}(\boldsymbol{\pi}_\alpha) \sim \text{Dir}(\boldsymbol{n}_\alpha^0)$, with concentration parameter vector $\boldsymbol{n}_\alpha^0 = (n_{\alpha 0}^0, \ldots, n_{\alpha C}^0)$. (34) A uniform prior (no prior knowledge) sets $n_{\alpha k}^0 = 1$ for all k. Prior information from an earlier $M \times D$ matrix R^0 (with $R_{\alpha i}^0$ as the category for the ith trial of the αth question) can be incorporated as:

$$n_{\alpha k}^0 = 1 + \sum_{i=1}^{D} \delta_{k, R_{\alpha i}^0}. \tag{6}$$

The Dirichlet prior is:

$$\mathcal{P}(\boldsymbol{\pi}_\alpha) = \frac{\Gamma(1 + C + D)}{\prod_{k=0}^{C} \Gamma(n_{\alpha k}^0)} \prod_{k=0}^{C} (\pi_{\alpha k})^{n_{\alpha k}^0 - 1}, \tag{7}$$

where $\sum_{k=0}^{C} n_{\alpha k}^0 = 1 + C + D$.

The normalization constant $\mathcal{P}(\boldsymbol{R}_\alpha)$ is:

$$\mathcal{P}(\boldsymbol{R}_\alpha) = \int_{\Delta} d\boldsymbol{p} \, \mathcal{P}(\boldsymbol{R}_\alpha | \boldsymbol{p}) \mathcal{P}(\boldsymbol{p}), \tag{8}$$

and since the Dirichlet is the conjugate prior, the posterior is $\mathcal{P}(\boldsymbol{\pi}_\alpha | \boldsymbol{R}_\alpha) \sim \text{Dir}(\boldsymbol{\nu}_\alpha)$, with $\boldsymbol{\nu}_\alpha = \boldsymbol{n}_\alpha + \boldsymbol{n}_\alpha^0$. The posterior distribution is:

$$\mathcal{P}(\boldsymbol{\pi}_\alpha | \boldsymbol{R}_\alpha) = \frac{\Gamma(T)}{\prod_{k=0}^{C} \Gamma(\nu_{\alpha k})} \prod_{k=0}^{C} (\pi_{\alpha k})^{\nu_{\alpha k} - 1}, \tag{9}$$

where $T \equiv \sum_{k=0}^{C} \nu_{\alpha k} = 1 + C + D + N$.

The moment generating function $\Phi(t) = \langle \exp(\bar{\pi} t) \rangle$ is:

$$\begin{aligned}
\Phi(t) &= \int_{\Delta} d\boldsymbol{\pi}_1 \cdots \int_{\Delta} d\boldsymbol{\pi}_M \exp(t\bar{\pi}) \prod_{\alpha=1}^{M} \mathcal{P}(\boldsymbol{\pi}_\alpha | \boldsymbol{R}_\alpha) \\
&= \prod_{\alpha=1}^{M} \int_{\Delta} d\boldsymbol{\pi}_\alpha \exp\left(\frac{t}{M} \sum_{k=0}^{C} w_k \pi_{\alpha k} \right) \mathcal{P}(\boldsymbol{\pi}_\alpha | \boldsymbol{R}_\alpha) \\
&= e^{t w_0} \prod_{\alpha=1}^{M} \int_{\Delta} d\boldsymbol{\pi}_\alpha \exp\left(t \sum_{k=1}^{C} s_k \pi_{\alpha k} \right) \mathcal{P}(\boldsymbol{\pi}_\alpha | \boldsymbol{R}_\alpha),
\end{aligned} \tag{10}$$

where $s_k \equiv (w_k - w_0)/M$, and $\pi_{\alpha 0} = 1 - \sum_{k=1}^{C} \pi_{\alpha k}$.

Each integral is the moment-generating function for a Dirichlet distribution, expressed via the confluent Lauricella hypergeometric function $\Psi^{[C]}$:

$$\Phi(t) = e^{tw_0} \prod_{\alpha=1}^{M} \Psi^{[C]}\left(\nu_{\alpha 1}, \ldots, \nu_{\alpha C}; T; ts_1, \ldots, ts_C\right), \tag{11}$$

where

$$\Psi^{[C]}\left(\nu_{\alpha 1}, \ldots, \nu_{\alpha C}; T; ts_1, \ldots, ts_C\right) = \sum_{m_1=0}^{\infty} \cdots \sum_{m_C=0}^{\infty} \frac{(\nu_{\alpha 1})_{m_1} \cdots (\nu_{\alpha C})_{m_C} (ts_1)^{m_1} \cdots (ts_C)^{m_C}}{(T)_m m_1! \cdots m_C!}, \tag{12}$$

and $(x)_n$ is the Pochhammer symbol.

The moments are:

$$\mu = \Phi'(0), \qquad \sigma^2 = \Phi''(0) - (\Phi'(0))^2. \tag{13}$$

Expanding $\Psi^{[C]}$ to $\mathcal{O}(t^2)$:

$$\begin{aligned} \Psi^{[C]} = 1 + \frac{t}{T} \sum_{j=1}^{C} \nu_{\alpha j} s_j + \frac{t^2}{2T(T+1)} \sum_{j=1}^{C} \nu_{\alpha j}(\nu_{\alpha j} + 1) s_j^2 \\ + \frac{t^2}{T(T+1)} \sum_{\ell=1}^{C} \sum_{m=\ell+1}^{C} \nu_{\alpha \ell} \nu_{\alpha m} s_\ell s_m + \mathcal{O}(t^3). \end{aligned} \tag{14}$$

Substituting into equation 11 and computing derivatives yields:

$$\begin{aligned} \mu &= w_0 + \frac{1}{MT} \sum_{\alpha=1}^{M} \sum_{j=0}^{C} \nu_{\alpha j}(w_j - w_0), \\ \sigma^2 &= \frac{1}{M^2(T+1)} \sum_{\alpha=1}^{M} \left\{ \sum_{j=0}^{C} \frac{\nu_{\alpha j}}{T}(w_j - w_0)^2 - \left(\sum_{j=0}^{C} \frac{\nu_{\alpha j}}{T}(w_j - w_0) \right)^2 \right\}. \end{aligned} \tag{15}$$

The algorithm summarizing this calculation is shown in Algorithm 1 in the main text.

B Proof of Equivalence of Bayesian and Average Rankings for Uniform Prior

For Bayesian estimators using a uniform prior (where $D = 0$, $T = 1 + C + N$, $\nu_{\alpha k} = 1 + n_{\alpha k}$), the expression for the mean μ from equation 15 simplifies as:

$$\begin{aligned} \mu &= w_0 + \frac{1}{M(1+C+N)} \sum_{\alpha=1}^{M} \sum_{j=0}^{C} (1 + n_{\alpha j})(w_j - w_0) \\ &= A + \frac{1}{M(1+C+N)} \sum_{\alpha=1}^{M} \sum_{j=0}^{C} w_j n_{\alpha j}, \end{aligned} \tag{16}$$

where the constant A is given by

$$A = \frac{1}{1+C+N} \sum_{j=0}^{C} w_j, \tag{17}$$

and $\sum_{j=0}^{C} n_{\alpha j} = N$. Here, μ relates to a naive weighted average accuracy a over the number of answers in each category,

$$a = \frac{1}{MN} \sum_{\alpha=1}^{M} \sum_{j=0}^{C} w_j n_{\alpha j}, \tag{18}$$

via

$$\mu = A + \frac{N}{1+C+N}a. \tag{19}$$

Note that in the binary case where $C = 1$, $w_0 = 0$, $w_1 = 1$, the value of a is the just regular average accuracy avg@N. For categorical cases, it is just a weighted generalization of avg@N.

Since A is constant across models and the prefactor $\frac{N}{1+C+N}$ is positive, we see that if $\mu > \mu'$, the corresponding values of a and a' from the two methods must always give the same ranking, $a > a'$. Additionally, in the limit of a large number of trials, $N \to \infty$, we see that $A \to 0$ and $\mu \approx a$, as expected.

This equivalence extends to uncertainty quantification. The relationship between the standard deviation of the average ($\sigma_{\text{avg@}N}$) and the Bayesian standard deviation ($\sigma_{\text{Bayes@}N}$ from equation 15) is

$$\sigma_{\text{avg@}N} = \frac{1+C+N}{N}\sigma_{\text{Bayes@}N}. \tag{20}$$

The Bayesian expression for $\sigma_{\text{Bayes@}N}$ is valid for all M and N, providing a reliable method to compute uncertainty in avg@N without relying on the Central Limit Theorem (CLT).

C Convergence

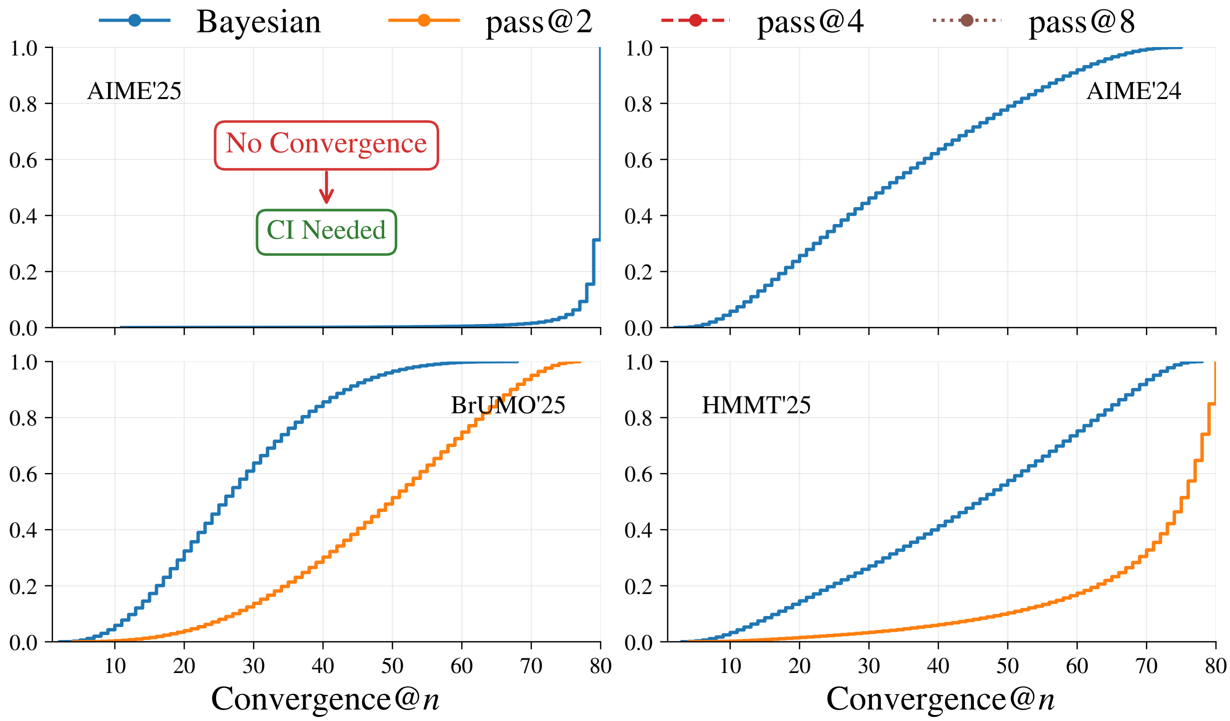

Figure 6: Complementing the PMFs in Fig. 1, these CDFs plot $P(k \leq N)$ for the convergence threshold k across AIME'24, AIME'25, HMMT'25, and BrUMO'25. Steeper and earlier rises indicate faster convergence. Bayes@N accumulates mass with fewer trials than Pass@2/4/8, and on AIME'24/'25 the Pass curves do not reach 1 by $N_{\max} = 80$. Greater convergence suggests that a confidence interval should be included in the evaluation tasks.

D Categorical

As we discussed in Sections 2.3 and 3.4, for each question $\alpha \in 1, \ldots, M$, every attempt yields base signals such as `has_box`, `is_correct`, `token_ratio`, `prompt_bpt`, `completion_bpt`, and verifier probabilities A, B, C for *correct*, *wrong*, and *invalid/off-task*. Using thresholds and Boolean criteria, each attempt is mapped into one of

$C + 1$ categories under a chosen schema (e.g., *Format Aware*, *Conf-Wrong Penalty*, *Efficiency-Adjusted*; Table 5). We instantiate categorical schemata and update posterior means via Dirichlet–multinomial inference, yielding metrics that preserve correctness while explicitly reflecting formatting, calibration, and efficiency.

Base signals All signals are directly obtainable from common LLM inference stacks such as Hugging Face transformers (68) and vLLM (69), via per-step scores/log-probs and termination metadata, and require no model-specific instrumentation; the verifier probabilities A, B, C are defined in D.

- **has_box**: 1 if a final boxed answer is present; else 0.
- **is_correct**: 1 if the answer is correct; else 0.
- **token_ratio**: completion tokens normalized by 32,768 (shorter is smaller).
- **repeated_pattern**: 0 if `finish_reason` is `stop`; else 1 (degenerate output).
- **prompt_bpt**: negative average prompt log-prob in bits/token (lower is better).
- **completion_bpt**: negative average completion log-prob in bits/token (lower is better).
- **compass_context_A**: verifier contextual probability of *correct*.
- **compass_context_B**: verifier contextual probability of *wrong*.
- **compass_context_C**: verifier contextual probability of *irrelevant/off-task*.

Reward models in evaluation. While reward models are most familiar from fine-tuning (e.g., RLHF), we use one as a *lightweight verifier* to supply per-attempt label probabilities for

$$\{A, B, C\} = \{\text{correct, wrong, invalid/off-task}\}$$

in evaluation. Concretely, we employ OpenCompass `CompassVerifier-3B` to produce (A, B, C) and then apply *contextual calibration* to obtain a more robust, prompt-stable label distribution: we evaluate next-token scores for the candidate labels at a fixed answer slot, subtract a content-free baseline logit b_y from the task logit s_y for each label y, and apply temperature scaling to yield calibrated probabilities

$$p(y \mid x) = \text{softmax}\left(\frac{s_y - b_y}{T}\right).$$

This helps us mitigate saturation and the entanglement of formatting and confidence seen with last-token probabilities, and improves probability calibration for downstream rubric scoring.

Selected categorical schema. We define 12 schemata (Table 5) using the rubric variables (Table 4) derived from the base signals; here are two illustrative definitions (the others follow analogously):

- **Format Aware**:

$$\text{cat} = \begin{cases} 0 & \text{invalid} \\ 1 & \text{wrong} \land \text{unboxed} \\ 2 & \text{wrong} \land \text{boxed} \\ 3 & \text{correct} \land \text{unboxed} \\ 4 & \text{correct} \land \text{boxed} \end{cases}$$

- **Conf-Wrong Penalty**:

$$\text{cat} = \begin{cases} 0 & \text{invalid} \\ 1 & \text{wrong}_{\text{high\_conf}} \\ 2 & \text{wrong} \land \text{low\_conf} \\ 3 & \text{correct} \end{cases}$$

Rubric weights $\mathbf{w}$ are chosen to reflect evaluation preferences. For example, *Format Aware* might use $[0, 0, 1, 2, 3]$ to mildly reward formatting when correct and slightly penalize confidently wrong (via schema choice); *Efficiency-Adjusted* can downweight verbose outputs among both correct and wrong categories.

- **Exact Match** Correctness only; ignores formatting, confidence, and length.
- **Format Aware** Rewards boxed, well-formatted answers; distinguishes boxed/unboxed even when wrong.

- **Conf-Calibrated** Penalizes *confidently wrong*; grades correct answers by confidence (low/mid/high).
- **OOD Robustness** Separates in-distribution vs. OOD prompts; checks correctness under both.
- **Strict Compliance** Requires boxed final answers; unboxed-correct is treated as non-compliant.
- **Conf-Wrong Penalty** Heavier penalty for wrong answers at high confidence; lighter when uncertain.
- **Verifier-Only** Uses verifier signals (A/B/C) alone to rank; model-agnostic probe of the verifier.
- **Format+Confidence** Balanced composite over (boxed/unboxed) × (low/high confidence) for both wrong and correct; emphasizes boxed, high-confidence correctness and penalizes confidently wrong.
- **Length-Robust** Isolates correctness irrespective of verbosity; does not penalize length.
- **Verifier Prob** Probes agreement with the verifier: flags wrong with high verifier A as inconsistent and distinguishes under/over-confidence on correct.
- **Efficiency-Adjusted** Rewards short, correct completions; penalizes verbose outputs (especially when wrong).
- **Concise High-Conf** Prefers concise, high-confidence correct answers; downweights verbose correctness.

Table 4: Rubric variables, decision formulas, and brief descriptions used to map each model attempt into discrete categories. Thresholds ($\tau_{\text{high}}, \tau_{\text{low\_wrong}}, \tau_{\text{prompt}}$) and length quantiles (len_p33, len_p66) are computed per dataset from observed bits-per-token and token-ratio statistics. Category 0 is reserved for invalid outputs (degenerate repetition or high verifier C), and A, B, C denote calibrated verifier probabilities for *correct*, *wrong*, and *off-task*, respectively.

Rubric variables	**Formula**	**Description**
invalid	$(\text{repeated\_pattern} = 1) \vee (C \geq 0.50)$	Category 0 reserved for invalid.
correct	$(\text{is\_correct} \geq 0.5)$	Boolean mask of correctness.
wrong	$(\text{is\_correct} < 0.5)$	Complement of correct.
high_conf	$(\text{completion\_bpt} \leq \tau_{\text{high}})$	Confidence proxy
low_conf	$(\text{completion\_bpt} > \tau_{\text{high}})$	Complement of high_conf.
wrong_high_conf	$\text{wrong} \wedge (\text{completion\_bpt} \leq \tau_{\text{low\_wrong}})$	Penalize confidently wrong.
ood	$(\text{prompt\_bpt} \geq \tau_{\text{prompt}})$	Out-of-distribution prompt.
ind	$(\text{prompt\_bpt} < \tau_{\text{prompt}})$	In-distribution prompt.
economical	$(\text{token\_ratio} \leq \text{len\_p33})$	Short completions.
moderate	$(\text{len\_p33} < \text{token\_ratio} \leq \text{len\_p66})$	Medium-length completions.
verbose	$(\text{token\_ratio} > \text{len\_p66})$	Long completions.
boxed	$(\text{has\_box} \geq 0.5)$	Answer is boxed.
unboxed	$(\text{has\_box} < 0.5)$	Answer is not boxed.
A_high	$(A \geq 0.6)$	Verifier confidence high.
τ_{high}	40th percentile of completion_bpt	
$\tau_{\text{low\_wrong}}$	60th percentile of completion_bpt among wrong items	
τ_{prompt}	90th percentile of prompt_bpt	
len_p33, len_p66	33rd and 66th percentiles of token_ratio	
corr_p33, corr_p66	33rd and 66th percentiles of completion_bpt correct items	

Fig. 7 summarizes aggregated results across tasks. The leader Qwen3-30B-A3B-Thinking ranks first under all selected schema, but the margin to rank 2 depends on the rubric (largest under *Conf-Wrong Penalty*, smallest under *Verifier-Only*. Mid-pack reorderings are rubric) sensitive: under *Verifier Prob*, *OpenThinker2-32B* edges *gpt-oss-20b_medium*; under calibration-heavy category (e.g., *Conf-Calibrated*, *Format+Confidence*), *gpt-oss-20b_high* overtakes *OpenThinker2-32B*; *OOD Robustness* narrows the gap between ranks 2 and 3. Several categories (*Format Aware*, *Length-Robust*, *Strict Compliance*) agree closely, indicating that once correctness is accounted for, formatting and length rarely flip top ranks. In contrast, calibration-focused categories emphasize and penalize confidently wrong behavior, and efficiency-oriented categories favor concision. The lower tier is stable across categories (*EXAONE-4.0-1.2B*, *OpenThinker3-1.5B*, *OpenReasoning-Nemotron-1.5B*, *Sky-T1-32B-Flash*, *DeepSeek-R1-Distill-Qwen-1.5B*), suggesting rubric choice primarily reshuffles the middle while preserving extremes. Overall, the categorical schema surfaces complementary facets—format compliance, calibration, efficiency, OOD robustness, and verifier alignment—making rubric-dependent differences explicit and enabling compute-efficient, uncertainty-aware comparisons aligned with evaluation goals.

Table 5: Definitions of the twelve categorical evaluation schemata used in our Dirichlet–multinomial framework. Each schema specifies decision rules over correctness, formatting (boxed/unboxed), confidence (via completion_bpt), prompt distribution (in-distribution vs. OOD), output economy (via token_ratio), and verifier signals (A, B, C). These rules map every attempt into $C{+}1$ discrete categories, enabling posterior means and credible intervals for any chosen weight vector $\mathbf{w}$.

Categorical Schema	**Rubric**
Exact Match	0 invalid; 1 wrong; 2 correct
Format Aware	0 invalid; 1 wrong $\wedge$ unboxed; 2 wrong $\wedge$ boxed; 3 correct $\wedge$ unboxed; 4 correct $\wedge$ boxed
Conf-Calibrated	0 invalid; 1 wrong $\wedge$ low_conf; 2 wrong_high_conf; 3 correct $\wedge$ low_conf; 4 correct $\wedge$ mid; 5 correct $\wedge$ high_conf
OOD Robustness	0 invalid; 1 ood $\wedge$ wrong; 2 ind $\wedge$ wrong; 3 ood $\wedge$ correct; 4 ind $\wedge$ correct
Strict Compliance	0 invalid; 1 wrong $\vee$ (correct $\wedge$ unboxed); 2 correct $\wedge$ boxed
Conf-Wrong Penalty	0 invalid; 1 wrong_high_conf; 2 wrong $\wedge$ low_conf; 3 correct
Verifier-Only	0 invalid; 1 high C; 2 high B; 3 A_high
Format+Confidence	0 invalid; 1 wrong $\wedge$ unboxed; 2 wrong $\wedge$ boxed $\wedge$ low_conf; 3 wrong $\wedge$ boxed $\wedge$ high_conf; 4 correct $\wedge$ unboxed $\wedge$ low_conf; 5 correct $\wedge$ unboxed $\wedge$ high_conf; 6 correct $\wedge$ boxed $\wedge$ low_conf; 7 correct $\wedge$ boxed $\wedge$ high_conf
Length-Robust	0 invalid; 1 wrong; 2 correct
Verifier Probe	0 invalid; 1 wrong $\wedge$ A_high; 2 wrong $\wedge$ $\neg$ A_high; 3 correct $\wedge$ $\neg$ A_high; 4 correct $\wedge$ A_high
Efficiency-Adjusted	0 invalid; 1 wrong $\wedge$ economical; 2 wrong $\wedge$ moderate; 3 wrong $\wedge$ verbose; 4 correct $\wedge$ economical; 5 correct $\wedge$ moderate; 6 correct $\wedge$ verbose
Concision-High-Conf	0 invalid; 1 wrong; 2 correct $\wedge$ verbose; 3 correct $\wedge$ moderate; 4 correct $\wedge$ economical; 5 correct $\wedge$ economical $\wedge$ high_conf

E Extended Related Work

The evaluation of LLMs in generative reasoning tasks, under test-time scaling (e.g., via repeated sampling(70)), has evolved to address the stochastic nature of inference and the need for robust measures of functional correctness. Early approaches relied on syntactic similarity metrics like BLEU (71) and CodeBLEU (72), which compare generated answers against reference solutions. However, these metrics often fail to capture semantic correctness in reasoning tasks, motivating metrics based on execution-validation or test-based validation (73; 72). This limitation has shifted focus toward functional evaluation, where the generated solution is assessed via a ground truth to verify correctness(73; 74). In this section, we review key functional metrics, focusing on those that leverage multiple samples to scale performance at inference time. These metrics form the basis to assess LLM capabilities but often overlook probabilistic uncertainty or consistency across samples, motivating our novel Bayesian framework.

The Pass@k metric, originally introduced by (73; 40) for evaluating LLMs trained on code. It measures the probability that at least one of k independently generated samples for a given problem passes all associated unit tests (i.e., by matching ground-truth answers or satisfying logical constraints), offering a practical estimate of a model's potential performance in solving a variety of complex tasks and problems. The unbiased estimator of Pass@k is computed as:

$$\text{Pass@}k = \mathbb{E}_{\text{problems}}\left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}\right], \tag{21}$$

where n is the total number of generated samples and c is the total number of correct solutions within the n trials. This estimator has smaller uncertainty in the limit of $n \gg k$, ensuring reliable approximations. However, due to computational costs, k is often comparable to n in practice, which can increase variance and weaken evaluation stability. The Pass@k metric has been adapted beyond code to evaluate LLMs in various tasks requiring verifiable correctness, such as math, logic, and general reasoning (74; 75; 76; 77).

Passˆk, introduced in (62), extends the Pass@k metric to capture both the potential performance and the consistency of LLMs in reasoning tasks, where evaluating the reliability and stability of generated solutions is crucial. Passˆk is defined as the probability that all k trials are correct:

$$\text{Pass\^{}}k = \mathbb{E}_{\text{problems}}\left[\frac{\binom{c}{k}}{\binom{n}{k}}\right], \tag{22}$$

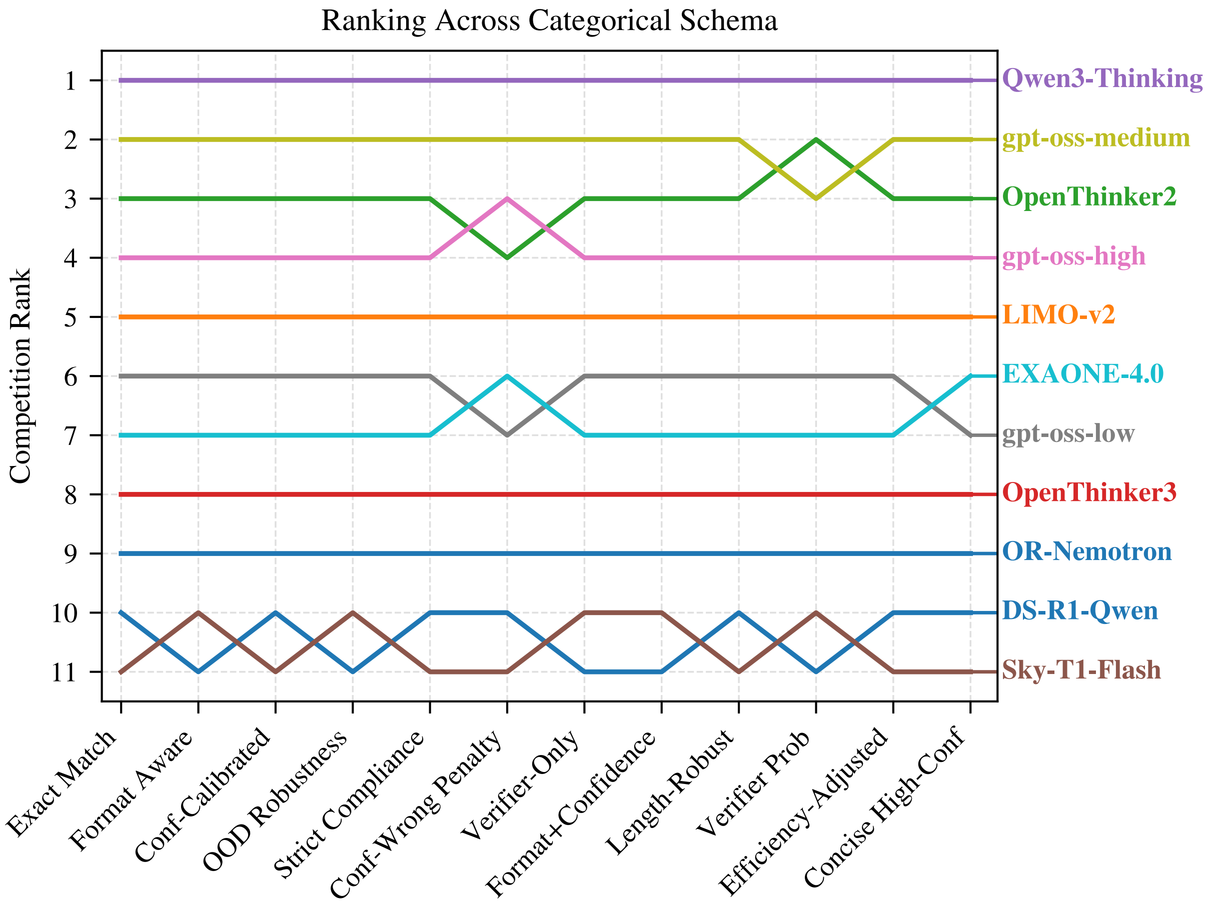

Figure 7: Competition ranks by model across selected categorical schema. Each column is a combination of base signals; lines indicate how a model's relative position shifts when the rubric changes.

where c and n retain the same meanings as in Pass@k. This metric assumes that all the trials are independent and uniformly distributed, approximating the binomial distribution with a hypergeometric distribution to account for sampling without replacement. By requiring all k samples to be correct, Pass^k provides a stringent measure of model consistency and stability.

To introduce flexibility, Liu et al. (26) proposed **G-Pass@$k_{\tilde{\tau}}$**, which incorporates a tolerance threshold $\tilde{\tau} \in (0.0, 1.0]$:

$$\text{G-Pass@}k_{\tilde{\tau}} = \mathbb{E}_{\text{problems}}\left[\sum_{j=\lceil \tau \cdot k \rceil}^{c} \frac{\binom{c}{j} \cdot \binom{n-c}{k-j}}{\binom{n}{k}}\right], \tag{23}$$

where $\lceil \tau \cdot k \rceil$ is the smallest integer greater than or equal to $\tau \cdot k$. This formulation allows up to $k - \lceil \tau \cdot k \rceil$ incorrect solutions, balancing the assessment of potential with consistency. As a special case, Pass@k corresponds to G-Pass@k_τ in the limit $\tau \to 0$.

Furthermore, Liu et al. (26) introduced **mG-Pass@k**, an interpolated metric that integrates G-Pass@k_τ over $\tau \in [0.5, 1.0]$:

$$\text{mG-Pass@}k = 2\int_{0.5}^{1.0} \text{G-Pass@}k_\tau d\tau \approx \frac{2}{k} \sum_{i=\lceil 0.5 \cdot k \rceil + 1}^{k} \text{G-Pass@}k_{i/k}, \tag{24}$$

providing a more comprehensive measure that jointly reflects performance potential and reasoning stability.

These extended metrics have been applied to mathematical reasoning benchmarks such as LiveMathBench, MATH, and AIME, where they reveal substantial performance degradation of LLMs under stricter stability requirements.

F Experiment setup and Reproducibility

F.1 Metrics

Kendall's Tau: Kendall's tau (τ) is a nonparametric rank correlation coefficient that quantifies the ordinal relationship between two ranked sets by evaluating the consistency in their orderings. For two rankings of n items, it examines all unique pairs (i, j) where $i < j$:

- A pair is *concordant* if the relative ordering of items i and j is the same in both rankings (both place i before j or vice versa).
- A pair is *discordant* if the relative ordering is different.
- Pairs with ties in either ranking are neither concordant nor discordant.

Define n_c as the number of concordant pairs, n_d as the number of discordant pairs, and $n_0 = n(n-1)/2$ as the total number of unique pairs. Let n_1 represent the number of tied pairs in the first ranking, and n_2 similarly for the second ranking. The two common variants are the following:

$$\text{Tau-a:} \quad \tau_a = \frac{n_c - n_d}{n_0} \qquad \text{(no adjustment for ties),} \tag{25}$$

$$\text{Tau-b:} \quad \tau_b = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}} \qquad \text{(adjusts for ties in both rankings).} \tag{26}$$

Tau-a assumes no ties and may underestimate correlation when ties occur. Tau-b, which corrects for ties, is better suited for datasets with equivalent rankings.

In our implementation, we use `scipy.stats.kendalltau` with its default variant='b', which computes τ_b efficiently and handles ties appropriately. The coefficient ranges from -1 (perfect disagreement) to $+1$ (perfect agreement), with 0 indicating no association. This metric provides a robust, distribution-free measure for comparing model performance rankings, particularly when ties reflect meaningful equivalences.

F.2 Models and Datasets

Datasets. We evaluate on four math–reasoning test sets: AIME'24 (35), AIME'25 (36), BrUMO'25 (38), and HMMT'25 (37). AIME is administered by the Mathematical Association of America and consists of two sets of 15 integer-answer problems; we use the 2024 and 2025 problem sets. For HMMT'25, we use the officially posted February 2025 contest set (algebra, geometry, number theory, and combinatorics). For BrUMO'25, we use the published 2025 problem sets from the tournament archive.

Models. Unless noted otherwise, we run each generator with the provider-recommended chat template (DeepSeek/Qwen style when unspecified) and identical decoding settings (below) to minimize template-induced variance. The model cohort includes: Sky-T1-32B-Flash (78) (reasoning-optimized "flash" variant tied to overthinking-reduction work), Qwen3-30B-A3B-Thinking-2507 (79) (Qwen3 series, reasoning variant), DeepSeek-R1-Distill-Qwen-1.5B (46) (distilled reasoning model), gpt-oss-20b (80) (OpenAI open-weight reasoning model; we use the default quantization, MXFP4. For prompting, we rely on OpenAI Harmony, which defines three levels of reasoning efforts. LIMO-v2 (81) (data-efficient reasoning fine-tuned on curated traces), EXAONE-4.0-1.2B (82) (hybrid non-reasoning/reasoning modes), OpenReasoning-Nemotron-1.5B (83; 84; 85; 86) (open-weight small reasoning model), OpenThinker2-32B (87) and OpenThinker3-1.5B (87) (trained on OpenThoughts2/3 data recipes). For verification we additionally use CompassVerifier-3B (88), a lightweight answer verifier suitable for outcome reward and equivalence checking.

Prompting. For most models, we follow the provider-recommended DeepSeek/Qwen-style prompt: *"Please reason step by step, and put your final answer within* `\boxed{}`*."* For `gpt-oss-20b`, we instead use the OpenAI Harmony prompt template, which provides three levels of reasoning effort. For `OpenReasoning-Nemotron-1.5B`, we adopt the task-specific prompt: *"Solve the following math problem. Make sure to put the answer (and only the answer) inside* `\boxed{}`*."*

F.3 REPRODUCIBILITY

- **Sampling setup.** All trials use top-p sampling with temperature 0.6, $p = 0.95$, batch size 1, and seeds 1234–1313. We perform $N = 80$ trials per dataset/model.
- **Serving stack.** Token generation is served with `vLLM` (PagedAttention) (69), and models are loaded in `bf16` unless the release requires MXFP4 (e.g., `gpt-oss`). We record log-probabilities for both the input prompt and generated tokens, and cap `max_tokens` at 32,768.
- **Verifier.** We use `CompassVerifier-3B` as a reward model. During evaluation, we leverage the model's scores on prompts generated by other models to create categorical schemas. We rely on the `Transformers` (68) and `Accelerate` (89) libraries. To maximize throughput, we enable FlashAttention kernels (22) and adopt the `DFloat11` format (90).
- **Hardware.** All runs execute on clusters with $8\times$ NVIDIA H200 (141GB).