TEST-TIME SCALING IS BOUNDED BY VERIFICATION, NOT GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time compute methods draw many candidate solutions and return one. What sampling makes available, coverage, is fixed by single-sample accuracy; what a user receives depends on selection, and selection is governed by other quantities. We prove that a verifier's AUC does not determine its best-of-k accuracy: with one correct candidate among k, a verifier of AUC a selects it with probability anywhere between a^{k-1} and a, so no verifier can recover a lone correct answer more often than its AUC, and the AUC needed for a fixed accuracy rises with the budget. For voting, whose success depends on whether the most common wrong answer outnumbers the right one, we show that the usual way of measuring difficulty, stratifying problems on the same samples used for evaluation, understates voting on hard problems in every model we test, enough to make it look worse than random for some, and overstates it on easy ones. With a held-out protocol across six models from three families, 1,400 problems and 32 samples each, real verifiers sit well inside the interval, so that their AUC leaves best-of-8 accuracy undetermined over tens of points, voting's held-out gains follow its error condition, and 32 samples predict coverage and reward-model accuracy at 128, including where selection levels off. On the 10,000 samples per problem released by Brown et al., the voting condition predicts the height of the voting plateau to within a point for 8B and 70B models. On hard problems, even a trained process reward model leaves 38–66% of the correct answers added by doubling the budget unselected. Sampling has outrun our ability to tell which sample is right.

1 INTRODUCTION

A dominant way to spend computation at inference time is to draw many candidate solutions and return one. Best-of-N selection with a verifier and self-consistency (Wang et al., 2023) both follow this pattern, and their returns are usually reported as $\mathrm{pass@}k$, the probability that at least one of k samples is correct (Chen et al., 2021). But $\mathrm{pass@}k$ measures what sampling makes *available*; a user receives only the candidate that a selector returns. The two diverge. Coverage keeps rising over four orders of magnitude of samples while majority voting and reward models level off near a hundred (Brown et al., 2024), and the accuracy of voting can even fall as samples are added (Chen et al., 2024). This paper explains why selection falls behind, shows that a standard way of measuring the shortfall produces effects that are not there, and tests the explanation by predicting behavior at budgets it has not seen.

A verifier's AUC does not determine what it can select. Coverage depends on single-sample accuracy alone (Proposition 1), so what sampling buys is fixed once accuracy is. Selection is governed by other quantities. For voting the governing quantity is known: voting converges to the correct answer exactly when that answer is more frequent than every wrong one (Condorcet, 1785; List & Goodin, 2001; Chen et al., 2024), a condition that cannot be checked without knowing the answer (Proposition 2). For verifiers we prove that the quantity by which they are usually compared, AUC, leaves selection underdetermined (Proposition 3): when one of k candidates is correct, a verifier of AUC a selects it with probability anywhere between a^{k-1} and a, and both ends are attained. No verifier can pick out a lone correct candidate more often than its AUC, and if its scores are concentrated, selecting it half the time takes an AUC of 0.91 at $k = 8$ and 0.98 at $k = 32$. The more one samples, the better the verifier must be for the added samples to be used.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

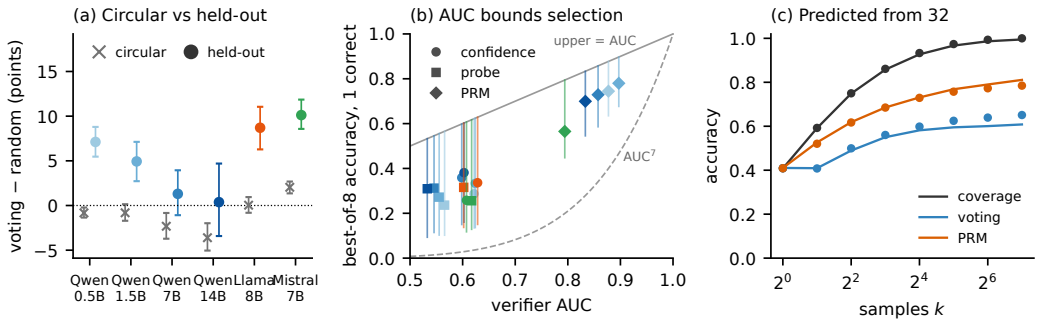


Figure 1: **(a)** The same data, evaluated two ways. Stratifying problems by the number of correct samples and evaluating voting on the same samples understates voting's gain on the hardest problems in every model, below zero for four of six models (circular); defining strata on held-out samples removes the bias. **(b)** Best-of-8 selection accuracy with one correct candidate, against the verifier's AUC, for every model and verifier. Bars show the interval of Proposition 3, whose upper end equals the AUC; it is wide, and verifiers sit well inside it. **(c)** Coverage, voting and process-reward-model accuracy at up to 128 fresh samples (points), predicted from 32 earlier samples (lines). Coverage and the reward model are predicted; voting is underpredicted.

Stratifying by difficulty on the same samples manufactures failures. Selection matters where correct samples are rare, so results are naturally reported by how many samples of a problem are correct. When that count is computed from the same samples on which voting is evaluated, the comparison is biased: it understates voting on hard problems, sometimes to the point of making it look worse than random choice, and overstates it on easy ones. On our data, this protocol understates voting's gain on the hardest problems by 3.6 to 8.7 points in every model, which is enough to make voting look worse than random choice for four of six models, and overstates it on easier problems. The bias shrinks as the number of samples grows and is negligible at a thousand (Section 4.5), so it arises precisely at practical budgets. An earlier analysis of our own data reached exactly this wrong conclusion. Every stratified result in this paper defines strata on one half of each problem's samples and measures outcomes on the other.

What we find. Across six instruction-tuned models from three families (0.5B to 14B parameters), 1,400 test problems and 32 samples per problem, with a trained process reward model as the strongest verifier:

- For every model and verifier, the interval of Proposition 3 spans 20 to 50 points of best-of-8 accuracy, and observed accuracies fall 26–55% of the way through it: AUC leaves selection largely undetermined. On hard problems, the process reward model leaves 38–66% of the correct answers that doubling the budget from four to eight samples makes available unselected; voting leaves 50–84%.
- Voting's gains on held-out samples follow the condition of Proposition 2 at matched accuracy, against a null that preserves accuracy and the average shape of errors (calibrated slope 0.61–0.93, $z = 3.7$–7.5, significant for 6 of 6 models after correction).
- Thirty-two samples per problem predict coverage and the reward model's best-of-k accuracy at 128 fresh samples, including the level at which the latter plateaus (81.1% predicted and 78.5% observed at $k = 128$); they underpredict voting by up to 4.3 points.
- On the 10,000 samples per problem released by Brown et al. (2024), the condition of Proposition 2 predicts the height at which voting levels off to within 0.9 points for Llama-3-8B and 70B on MATH and GSM8K (43.8% predicted and 43.8% observed for Llama-3-8B on MATH, while coverage reaches 97%).

Contributions. (i) A tight bound relating a verifier's AUC to its best-of-k selection accuracy, which gives a quantitative account of why reward-model selection plateaus. (ii) The identification of a circularity in difficulty-stratified evaluation, and a held-out protocol with a calibrated null that

removes it. (iii) Held-out evidence across three model families, including out-of-sample prediction of large-budget behavior from small samples, and a reanalysis of the released samples of Brown et al. (2024) that predicts the voting plateau they reported.

2 COVERAGE AND SELECTION

Fix a problem and a decoding configuration. The model induces a distribution over solutions, and drawing k samples means drawing k solutions independently from it. Each solution yields an answer. Let p be the probability that a sample's answer is correct, and let $q_1 \geq q_2 \geq \dots$ be the probabilities of the distinct wrong answers, so that $p + \sum_j q_j = 1$. Because solutions have bounded length, only finitely many answers, say M, have positive probability. *Coverage* at budget k is the probability that at least one of k samples is correct. *Selection accuracy* is the probability that a procedure which must return one of the k samples returns a correct one.

2.1 COVERAGE DEPENDS ON ACCURACY ALONE

Proposition 1. *Coverage at budget k equals $1 - (1-p)^k$. For a task whose problems have accuracies distributed as F, coverage is $C(k) = \mathbb{E}_{p\sim F}\left[1-(1-p)^k\right]$. In particular, once p is known, no statistic of the samples carries information about coverage.*

Proof. Correctness indicators of independent samples are i.i.d. Bernoulli(p), so the probability that none is correct is $(1-p)^k$. Averaging over problems gives $C(k)$. Given p, the event that some sample is correct has a probability fixed by p, hence is independent of any further statistic. □

The proposition is elementary, but it constrains what diversity can explain. A measure of how spread out the samples are can correlate with coverage only by correlating with p. Because $p \mapsto 1-(1-p)^k$ is strictly concave for $k \geq 2$, Jensen's inequality gives $C(k) \leq 1-(1-\mathbb{E}[p])^k$: coverage depends on the shape of F, not only on its mean. Aggregate comparisons across models therefore compare different populations of problems, which is why we compare selectors within strata of matched accuracy (Section 3).

2.2 VOTING DEPENDS ON THE MOST COMMON ERROR

Majority voting returns the most frequent answer among the k samples, breaking ties uniformly at random. Random choice returns one sample uniformly, and is correct with probability p for every k.

Proposition 2. *Let $\Delta = p - q_1$. If $\Delta > 0$, majority voting fails with probability at most $(M-1)\exp(-k\Delta^2/2)$. If $\Delta < 0$, it succeeds with probability at most $\exp(-k\Delta^2/2)$. Hence, as k grows, voting becomes more accurate than random choice when the correct answer is more frequent than every wrong answer, and less accurate when some wrong answer is more frequent than the correct one, whatever the value of p.*

Proof. Let N_c and N_j count the samples with the correct answer and with the j-th wrong answer. Voting can fail only if $N_j \geq N_c$ for some j. The variables $X_i = \mathbf{1}\{\text{sample } i \text{ correct}\} - \mathbf{1}\{\text{sample } i \text{ gives answer } j\}$ are i.i.d., take values in $[-1,1]$, and have mean $p - q_j \geq \Delta$. By Hoeffding's inequality, $\Pr[N_c - N_j \leq 0] \leq \exp(-k(p-q_j)^2/2) \leq \exp(-k\Delta^2/2)$ when $\Delta > 0$, and a union bound over the $M-1$ wrong answers gives the first claim. If $\Delta < 0$, voting can succeed only if $N_c \geq N_1$; applying the same inequality to $N_1 - N_c$, whose mean is $-\Delta k > 0$, gives the second. □

The condition is not new: it is a plurality form of the Condorcet jury theorem (Condorcet, 1785; List & Goodin, 2001), and Chen et al. (2024, Lemma 1) use the same quantity, $q_1 - p$, as a difficulty indicator for voting on tasks with few possible answers. We state it with an explicit finite-budget bound because our budgets are small and our answer spaces large. Two consequences matter here. First, whether voting helps is decided by the sign of $p - q_1$, which Proposition 1 shows to be irrelevant to coverage. Second, that sign depends on which answer is correct, so it cannot be estimated from unlabeled samples: voting cannot detect its own failures. What the proposition does not settle is

whether the condition can be measured reliably from the tens of samples used in practice; Section 3 shows that the obvious way of measuring it is biased.

2.3 Verification depends on the tail, not the AUC

Our main theoretical result concerns verifiers. A verifier assigns a score to each candidate and the candidate with the highest score is returned. Suppose that among k candidates, m are correct, with $1 \le m < k$. Let correct candidates have i.i.d. scores with distribution F_+, incorrect candidates i.i.d. scores with continuous distribution F_-, all independent, and let $a = \Pr[S_+ > S_-]$ be the verifier's AUC on the problem.

Proposition 3. *The probability that the verifier selects a correct candidate satisfies*

$$a^{\,k-m} \;\le\; \Pr[\textit{correct selected}] \;\le\; 1-(1-a)^m, \tag{1}$$

and both bounds are attained. For a single correct candidate ($m = 1$), selection accuracy lies in $[a^{k-1}, a]$.

Proof. Let $V_i = F_-(S_{+,i})$ for the correct candidates; then $V_i \in [0,1]$ and $\mathbb{E}[V_i] = a$. Conditional on the correct scores, all $n = k - m$ incorrect scores fall below the largest correct score with probability $F_-(\max_i S_{+,i})^n = (\max_i V_i)^n$, so $\Pr[\text{correct selected}] = \mathbb{E}[(\max_i V_i)^n]$. For the lower bound, $\mathbb{E}[(\max_i V_i)^n] \ge \mathbb{E}[V_1^n] \ge a^n$ by Jensen's inequality, since $x \mapsto x^n$ is convex. For the upper bound, $\mathbb{E}[(\max_i V_i)^n] \le \mathbb{E}[\max_i V_i] = \int_0^1 (1-(1-h(t))^m)\,dt$ with $h(t) = \Pr[V_1 > t]$ and $\int_0^1 h(t)\,dt = a$; the integrand is concave in h, so Jensen's inequality bounds the integral by $1-(1-a)^m$. The lower bound is attained when every V_i equals a, the upper when each V_i is 0 or 1. □

The proof is elementary; what has not been drawn out, to our knowledge, is its consequence for test-time scaling. AUC measures how often a correct candidate outranks an incorrect one in a single pair; it is the $k = 2$ case of the quantity that matters. Best-of-k asks the correct candidate to outrank every incorrect one at once, which depends on how the verifier's errors are distributed in the upper tail, and AUC does not constrain that. Three consequences follow.

- *AUC is a ceiling for rare answers.* No verifier selects a lone correct candidate more often than its AUC, however its scores are shaped.
- *The required AUC grows with the budget.* If the scores are concentrated, as the lower bound describes, reaching selection accuracy s requires $a \ge s^{1/(k-m)}$; for $s = 0.5$ and $m = 1$ this is 0.79 at $k = 4$, 0.91 at $k = 8$ and 0.98 at $k = 32$.
- *AUC compresses differences between verifiers.* Because the bounds are convex and concave powers of a, a small difference in AUC can correspond to a large difference in best-of-k accuracy, and the ordering of two verifiers by AUC need not match their ordering at large k.

3 Measuring Selection Without Circularity

Models, problems and sampling. We use Qwen2.5-Instruct at 0.5B, 1.5B, 7B and 14B parameters (Qwen Team, 2024), Llama-3.1-8B-Instruct (Llama Team, 2024) and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023). The problem set is frozen before any generation and shared by all models: 200 problems from each of the five levels of the MATH test set, drawn with a fixed seed across all seven subjects (Hendrycks et al., 2021), and 200 problems each with short and long reference solutions from the GSM8K test set (Cobbe et al., 2021), 1,400 problems in all. Each model receives its own chat template and the instruction *"Please reason step by step, and put your final answer within \boxed{}"*; we draw 32 samples per problem at temperature 0.8 and nucleus 0.95 with up to 2,048 new tokens (Kwon et al., 2023). At most 4.9% of samples reach the limit in any model. Samples that reach the limit are almost all degenerate: for Llama-3.1-8B on the hardest MATH level, where 15.5% reach it, only 2% of them contain a final answer, and their endings are more repetitive than those of samples that finish (median compression ratio 0.27 versus 0.38). We score them as incorrect; removing them does not change the voting result (Appendix C). Answers are scored with Math-Verify (Hugging Face, 2025), which agrees with an exact-match scorer on 93.3–96.9% of samples; for voting, all correct answers form one class and incorrect answers are grouped by normalized value.

Selectors. We compare random choice, majority voting with uniform tie-breaking, and three verifiers: model confidence (mean token log-probability), a linear probe on text embeddings (Wang et al., 2022) trained on other problems by two-fold cross-fitting, and the process reward model Qwen2.5-Math-PRM-7B (Zhang et al., 2025), aggregated by the minimum over steps. The reward model scores all samples of the first 100 problems per task for the Qwen models and of the first 50 for Mistral. The oracle returns a correct sample whenever one is present; its accuracy is coverage.

The circularity. Let a problem's n samples contain c correct ones. If c is used both to assign the problem to a difficulty stratum and as the pool on which voting is evaluated, stratum and outcome are dependent: problems with small c are over-represented by draws in which the correct answer happened to be scarce relative to its true frequency, and voting on those draws is handicapped by exactly that scarcity. Random choice is not handicapped in the same way, because its accuracy on the pool is c/n by construction. The bias runs the other way for large c. This is regression to the mean, a familiar hazard whenever units are selected on a noisy measurement of the quantity being studied; what is specific here is that the selected quantity is also the pool from which voting draws, so the bias falls on voting and not on random choice. It vanishes as n grows, which is why it does not arise when difficulty is estimated from a thousand calls (Chen et al., 2024), but it is large at the tens of samples used in practice (Sections 4.2 and 4.5).

Held-out protocol. We split each problem's samples into halves A and B by a fixed hash, define strata and all estimated quantities on A, and evaluate every selector on subsets of size $k \leq 16$ drawn from B. Estimation noise in A moves problems into the wrong stratum, which can only shrink differences between strata, so the protocol is conservative. To use all problems rather than a few strata, we test Proposition 2 by regressing the held-out voting gain on the gain predicted from A's answer distribution within strata of equal accuracy. Because A still carries information about true accuracy beyond its stratum, we calibrate this slope against a null in which each problem keeps its accuracy but its error distribution is replaced by the average shape of errors among problems with the same accuracy, and report the slope minus its null mean.

Verifier protocol. Verifiers score each sample independently, so they do not share voting's dependence on the pool. For each problem we compute the verifier's AUC and, by the formula in the proof of Proposition 3 applied to the empirical score distributions, the exact probability that a correct candidate wins among m correct and $k - m$ incorrect candidates drawn with replacement. Bounds are computed per problem and then averaged, since they are nonlinear in the AUC.

Prediction at larger budgets. For 144 problems on which Qwen2.5-7B-Instruct has between 1 and 24 correct samples out of 32, we draw 128 further samples with different random seeds, independent of the first 32. From the first 32 alone we predict coverage, voting and reward-model accuracy for k up to 128, and compare with the fresh samples. Selecting problems by their count of correct samples biases naive estimates towards the extremes, so accuracy is estimated with a beta prior fitted to all 1,400 problems; wrong answers seen once are treated as always new; and reward-model scores are modeled as normal on the logit scale, with the separation between correct and incorrect scores shrunk toward its population mean by empirical Bayes. We validated this procedure on synthetic data with known ground truth before looking at the fresh samples (Appendix F).

Statistics. Intervals are 95% percentile intervals from 2,000 bootstrap resamples of problems. Decisions fixed before the corresponding data were examined, and the two that changed after we saw partial results, are listed in Appendix B.

4 Results

4.1 A verifier's AUC does not determine what it can select

The interval is wide, and real verifiers sit inside it. Proposition 3 holds for any score distributions, so applied to each problem's empirical scores it cannot be violated; we checked 22,484 verifier–problem cases as a test of the computation, not of the theory. The empirical question is how much the bound constrains real verifiers, and the answer is: little. Across models and verifiers, the interval for one correct candidate among eight spans 20 to 50 points of accuracy (for model confidence, 14 to 61

Table 1: Main results. Acc.: single-sample accuracy (%). AUC: per-problem AUC of model confidence and of the process reward model (PRM). Best-of-8: PRM accuracy with one correct candidate among eight, with the interval of Proposition 3 (%). Unselected: share of the correct answers added by doubling the budget from 4 to 8 on hard problems that the selector does not return (%). β: calibrated held-out slope for Proposition 2 at $k = 8$ (z against the null). Circ./held: voting minus random on the hardest stratum, circular versus held-out protocol (points). – : not scored.

		AUC		Best-of-8	Unselected			
Model	Acc.	conf.	PRM	PRM [bounds]	PRM	vote	β (z)	Circ./held
Qwen 0.5B	37.7	0.62	0.88	74 [63, 88]	37.9	60.7	0.87 (7.5)	-0.8 / +7.1
Qwen 1.5B	61.8	0.62	0.90	78 [68, 90]	40.5	66.9	0.83 (7.3)	-0.8 / +4.9
Qwen 7B	82.6	0.60	0.86	73 [59, 86]	58.8	83.6	0.61 (3.7)	-2.3 / +1.3
Qwen 14B	87.0	0.60	0.83	70 [55, 83]	66.4	83.3	0.86 (4.7)	-3.6 / +0.4
Llama 8B	61.2	0.63	–	–	–	52.8	0.93 (7.0)	+0.0 / +8.7
Mistral 7B	26.7	0.61	0.79	57 [45, 79]	54.1	50.5	0.87 (7.1)	+2.0 / +10.1

percent on average), and observed accuracies fall 26–55% of the way from its lower to its upper end (Figure 1b). Knowing a verifier's AUC therefore says little about how it will select. What the proof supplies beyond the interval is a representation of best-of-k accuracy through the upper tail of the score distributions, which Section 4.4 uses to predict accuracy at budgets not yet sampled.

A trained verifier is far better, and still leaves most added answers unused. The process reward model reaches an AUC of 0.79–0.90 and selects the lone correct candidate among eight in 57–78% of cases, against twelve and a half percent for random choice (Table 1). Yet on hard problems it leaves 38–66% of the correct answers that doubling the budget from four to eight samples makes available unselected, and voting leaves 50–84%. Coverage rises; what is selected does not keep pace.

AUC understates the difference between verifiers. The process reward model's AUC exceeds model confidence's by 0.20–0.27, but its best-of-eight accuracy with one correct candidate exceeds it by 0.31–0.46, in every model the reward model scores (Appendix Figure 3). A comparison by AUC compresses exactly the difference that matters for test-time scaling. Across all pairs of verifiers and all models, the ordering by AUC and by best-of-k accuracy never reversed significantly; the compression is a matter of scale, not of rank.

4.2 Stratifying on the same samples manufactures failures

The circular protocol understates voting where it matters most. Among problems whose correct samples make up at most an eighth of the pool, the circular protocol reports a smaller gain from voting than the held-out protocol in every model, by 3.6 to 8.7 points (Figure 1a). For four of six models the understatement turns a gain into an apparent loss: voting looks worse than random choice (-3.6 to -0.8 points) when held-out samples show a gain ($+0.4$ to $+7.1$ points). On easier problems the bias runs the other way in every model, inflating voting's gain (held-out $+18.8$ to $+32.3$ points; circular $+23.4$ to $+39.9$ points). The circular protocol exaggerates how strongly voting depends on difficulty in both directions, and whether it produces an outright reversal depends on how large the true gain is relative to the bias.

Comparisons across models need matched difficulty. The same issue affects comparisons between models. The reward model's AUC is lower on the 14B model's samples than on the 0.5B model's, which suggests that stronger generators make errors that are harder to detect. On problems that both models find neither trivial nor impossible, the difference reverses (Appendix D): the apparent trend came from the stronger model retaining only harder problems.

4.3 Voting follows the error condition on held-out samples

The predicted gain predicts the realized gain. Within strata of equal estimated accuracy, the voting gain predicted from half of each problem's samples predicts the gain realized on the other half, with a calibrated slope of 0.61–0.93 at $k = 8$ ($z = 3.7$–7.5 against 200 null replicates; significant

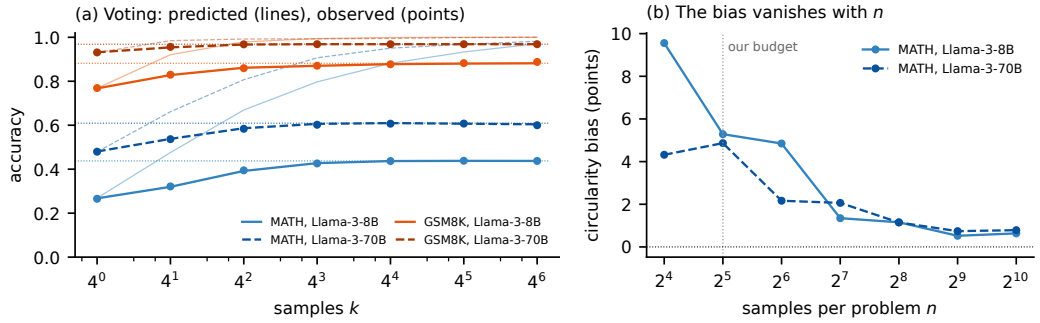


Figure 2: Reanalysis of the released samples of Brown et al. (2024), 10,000 per problem. **(a)** Majority voting predicted from one half of each problem's samples (lines) and observed on the other half (points), with coverage (thin lines); dotted lines mark the share of problems whose correct answer is more frequent than every wrong one, the limit given by Proposition 2. **(b)** The circularity bias of Section 4.2, for pools of n samples per problem.

for 6 of 6 models after Holm correction). The null slope is positive, so without calibration the effect would be overstated. The result holds with the exact-match scorer, on MATH alone and with truncated samples removed; on GSM8K alone it fails for the strongest models, which leave too few problems with both correct and incorrect samples to test (28 of 30 conditions; Appendix C).

On hard problems, voting recovers few of the available correct answers. Given that a correct sample is present, voting returns one in 20–37% of draws of eight on the hardest stratum, against 20–22% for random choice, far from the certainty that the pool allows. The condition of Proposition 2 explains why: on these problems the most common answer is usually wrong, and additional samples reinforce it rather than the correct one.

4.4 Thirty-two samples predict where selection levels off

The reward model's plateau was predicted before it was observed. From 32 samples per problem, we predicted coverage, voting and reward-model accuracy at up to 128 fresh samples on 144 hard problems, and then drew the fresh samples (Figure 1c). At $k = 128$ they show coverage 100%, reward-model selection 78% and voting 65%: coverage is complete while selection has leveled off, reproducing the plateau reported by Brown et al. (2024). The prediction for the reward model, which rests on Proposition 3's representation of best-of-k accuracy through the upper tail of the score distributions, matched at every budget (81.1% predicted and 78.5% observed at $k = 128$), as did coverage; 6 of 9 preregistered checks passed.

Voting was underpredicted. The prediction for voting fell below the observed accuracy by up to 4.3 points at $k \geq 32$ and failed the preregistered criterion. One contributing factor is visible in the data: the frequency of the most common wrong answer, estimated from 32 samples, was 29.6% on average but 27.2% in the fresh samples, and it was overestimated for 71% of 142 problems. An overestimated rival makes voting look weaker than it is, in the direction of the shortfall; we have not established that it accounts for all of it. Estimating the most common error from few samples is the same difficulty that underlies the circularity of Section 4.2.

4.5 The voting plateau in 10,000 samples per problem

Proposition 2 predicts the height of the plateau. Brown et al. (2024) released 10,000 samples per problem from Llama-3-8B-Instruct and Llama-3-70B-Instruct on 128 MATH and 127 GSM8K problems, the data on which they observed that voting levels off. We estimated each problem's answer distribution from half of its samples and compared the prediction with voting on the other half. By Proposition 2, voting converges to the share of problems whose correct answer is more frequent than every wrong one; this single number matches the observed accuracy at $k = 4{,}096$ to within 0.9 points in all four model–task pairs (43.8% predicted and 43.8% observed for Llama-3-8B

on MATH, while coverage reaches 97%; Figure 2a, Table 6). The finite-budget prediction met the preregistered criterion in four of four pairs. The plateau they described is the fraction of problems on which the most common answer is right.

The circularity bias is a small-sample effect. Drawing pools of n samples per problem from the same data, the circular protocol understates voting's gain on the hardest problems by 4.3 to 9.6 points at $n = 16$ and 4.9 to 5.3 at $n = 32$, the budget of our own study, falling to 0.6 to 0.8 at $n = 1{,}024$ (Llama-3-8B on MATH and Llama-3-70B on MATH; Figure 2b). This is why the bias does not appear when difficulty is estimated from a thousand samples (Chen et al., 2024) and why it matters at the budgets used in practice.

5 Related Work

Scaling inference compute. Self-consistency selects the most frequent answer among sampled reasoning chains (Wang et al., 2023). Brown et al. (2024) show that coverage grows over four orders of magnitude of samples, while majority voting, reward-model best-of-N and reward-weighted voting plateau near a hundred samples; they describe the plateau without explaining it. Chen et al. (2024) show that voting can be non-monotone in the number of calls and explain this with a difficulty indicator, the gap between the most frequent wrong answer and the correct one, estimated from up to a thousand calls per query on tasks with few possible answers. Work on compute-optimal inference compares sampling, search and revision under a fixed budget (Snell et al., 2025; Wu et al., 2025). We give a bound that accounts for the reward-model plateau, show that the difficulty indicator cannot be measured by stratifying on the same few samples used for evaluation, and test both explanations on held-out samples, including their own released samples.

Voting theory. Majority voting converges to the correct answer exactly when it is the most likely one, a plurality form of the Condorcet jury theorem (Condorcet, 1785; List & Goodin, 2001).

Verifiers. Outcome verifiers (Cobbe et al., 2021) and process reward models (Lightman et al., 2024; Zhang et al., 2025) are the standard way to select among samples, and imperfect verifiers are known to limit the gains from resampling (Stroebl et al., 2024). Verifiers are commonly compared by ranking metrics such as AUC; Proposition 3 shows that AUC bounds best-of-k accuracy only loosely, and tightly only from above when correct answers are rare.

6 Limitations

Our tasks have automatically checkable answers; open-ended generation has no oracle, so coverage is undefined there. We use one sampling temperature. Held-out comparisons use budgets up to 16 and the prediction experiment up to 128, on one model. Proposition 3 assumes that candidates of the same problem are scored independently, which a verifier that sees all candidates at once would violate. The reward model scores a subset of problems for five of the six models and is itself trained on outputs of one of our model families; on problems common to both, its AUC is 0.08–0.20 higher on Qwen samples than on Mistral samples, so its absolute accuracy should not be read as family-independent. Our own generations use models from 0.5B to 14B parameters; the reanalysis of released samples extends the voting results to 70B but carries no verifier scores. On GSM8K the strongest models leave too few problems with both correct and incorrect samples to test Proposition 2. In the check that the two halves of each problem's samples are exchangeable, one of six models showed a small difference between halves (0.7 points in single-sample accuracy) with intervals just excluding zero; the halves are split by a hash that is independent of generation order, and the difference is small relative to the held-out effects we report (Appendix E). Proposition 2 is known; our contribution concerning voting is its measurement.

7 Conclusion

Coverage is fixed by how often a model is right, so the extra samples of test-time scaling make correct answers available at a rate no property of the samples can change. What a user receives depends on

selection. Voting fails where the most common error outnumbers the truth, which it cannot see; a verifier fails in the upper tail of its errors, which its AUC does not measure. Both failures are easy to misjudge, because measuring difficulty on the samples being judged manufactures failures of its own. Sampling has outrun our ability to tell which sample is right.

AI USE STATEMENT

We used generative AI tools to assist with writing and editing prose, with generating and debugging experimental and analysis code, with retrieving related work, with proposing candidate hypotheses and experimental designs, and with generating the synthetic data used to validate the analysis code. The authors evaluated every proposal independently and rejected several after statistical control, including the circular stratified comparison described in Appendix B. All reported results were produced by code the authors executed and inspected, and every number and table in the paper is generated from logged runs by a script rather than transcribed by hand. The authors take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

Appendix A describes the models, the frozen problem set and its selection, prompting, sampling, scoring and every verifier. The problem set with its content hash and all generation, scoring and analysis code are provided as supplementary material; all sampled solutions and scores will be released with the paper. Generation uses fixed per-problem seeds, the held-out split is a fixed hash of problem identifiers, and every number, table and figure in the paper is produced from the logged results by the scripts `fill_numbers.py` and `make_figures.py` rather than transcribed by hand.

ETHICS STATEMENT

This work analyzes the outputs of publicly released language models on public mathematics benchmarks and the publicly released samples of Brown et al. (2024). It involves no human subjects or personal data. Its findings concern how test-time computation is evaluated; we do not foresee direct harmful applications.

REFERENCES

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.

Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. Are more LLM calls all you need? towards the scaling properties of compound AI systems. In *Advances in Neural Information Processing Systems*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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

Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Imprimerie Royale, Paris, 1785.

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. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Hugging Face. Math-verify: A robust mathematical expression evaluation system. https://github.com/huggingface/Math-Verify, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

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 *Symposium on Operating Systems Principles*, 2023.

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

Christian List and Robert E. Goodin. Epistemic democracy: Generalizing the Condorcet jury theorem. *Journal of Political Philosophy*, 9(3):277–306, 2001.

Llama Team. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Shmuel Nitzan and Jacob Paroush. Optimal decision rules in uncertain dichotomous choice situations. *International Economic Review*, 23(2):289–297, 1982.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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

Benedikt Stroebl, Sayash Kapoor, and Arvind Narayanan. Inference scaling fLaws: The limits of LLM resampling with imperfect verifiers. *arXiv preprint arXiv:2411.17501*, 2024.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

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. In *International Conference on Learning Representations*, 2023.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *International Conference on Learning Representations*, 2025.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025.

A Experimental Details

Models. Qwen/Qwen2.5-{0.5B, 1.5B, 7B, 14B}-Instruct, meta-llama/Llama-3.1-8B-Instruct and mistralai/Mistral-7B-Instruct-v0.3, all in bfloat16.

Problems. MATH test problems are pooled across the seven subjects at each level, ordered by a hash of the problem text and shuffled with a fixed seed; the first 200 per level are kept. GSM8K test problems are split by the number of lines in the reference solution (at most three for short, at least five for long), and 200 of each are kept the same way. The resulting set is frozen in a single file whose content hash is recorded, and every model uses exactly this file.

Prompting and sampling. Each problem is given as a single user turn, followed by the instruction quoted in Section 3, with the model's default chat template. For Mistral the template is passed as token identifiers: in a discarded run, passing it as a string caused its control tokens to be read as text, and 18% of samples continued past the answer with an invented user turn. We sample with vLLM at temperature 0.8 and nucleus 0.95, 32 samples per problem, at most 2,048 new tokens and a context of 4,096 tokens. Each problem has a fixed seed derived from its identifier; the 128 additional samples of Section 4.4 use a different base seed. Generation and reward-model scoring ran on NVIDIA H100 GPUs, about seven GPU-hours in total; embeddings and all analyses ran on a single consumer GPU.

Scoring and answer classes. Math-Verify decides correctness. A second scorer extracts the last boxed expression, or failing that the final number, and compares normalized strings and numerical values; it is used for the robustness check in Appendix C. For voting, every sample judged correct belongs to one class; incorrect answers are grouped after removing spaces, commas, dollar signs and `\text`, converting simple fractions and decimals to a common numerical form, and lowercasing.

Verifiers. *Confidence*: mean (and minimum) log-probability of the sampled tokens, recorded during generation. *Probe*: logistic regression on e5-base-v2 embeddings of the solution (mean pooling, at most 512 tokens), trained on 4,000 correct and 4,000 incorrect samples from the other half of the problems, split by a hash of the problem identifier. *Process reward model*: Qwen2.5-Math-PRM-7B, run with the library versions current at its release (transformers 4.48.3, torch 2.5.1) and the input format of its model card, with steps separated at blank lines. Step scores are aggregated by their minimum; Table 5 compares other aggregations. The reward model scores all samples of the first 100 problems per task for the Qwen models and of the first 50 for Mistral; it was not run on Llama because of the compute budget.

Held-out analyses. Halves are formed by a permutation seeded by a hash of the split index and the problem identifier. The test of Proposition 2 averages over five splits. Its null keeps each problem's number of correct samples and replaces the frequencies of its wrong answers by the average frequency profile of problems with the same number of correct samples; the per-budget table uses 30 null replicates and the primary test 200.

B Decisions and Changes

We record the analytical decisions that could have been made after seeing results, and when they were made.

- *Primary test of Proposition 2.* We first compared problems with $\hat{q}_1 > \hat{p}$ and $\hat{q}_1 < \hat{p}$ within strata. On the first model analyzed (Qwen2.5-7B), a simulation showed that this comparison is biased when q_1 correlates with p, in either direction. We replaced it with the calibrated regression of Section 3 before analyzing any other model.
- *Headline budget for unselected answers.* We had planned to report the doubling from 8 to 16 samples. Synthetic checks showed that its denominator, the gain in coverage, is too small on held-out halves of 16 samples to estimate, so we switched to 4 to 8 before computing it with the reward model. We had already seen the corresponding values without the reward model.
- *Rank reversals.* We fixed in advance that a reversal between AUC and best-of-k rankings would be claimed only if significant in two or more models. None was, and we make no such claim.
- *Scale trend of the reward model.* We fixed in advance that a trend would be claimed only if it held on problems common to both models in two of three comparisons. It did not.
- *Prediction experiment.* The prediction procedure, its criterion (predicted and observed within a 95% interval or two points at $k \geq 32$) and its problem selection were fixed and validated on synthetic data before the fresh samples were drawn. Coverage and the reward model met it at every budget; voting did not, and we report that failure rather than adjust the procedure after seeing it.
- *Reanalysis of released samples and likelihood-ratio voting.* Both analyses, their criteria and the rule for excluding settings with too few hard problems were fixed and checked on

synthetic data before we looked at the results. The reanalysis met its criteria; likelihood-ratio voting did not (Appendix I).

- *Circular analysis.* An earlier version of this study, with shorter generations, a different problem set and stratification on the same samples, concluded that voting is worse than random on hard problems. All results in this paper come from a regeneration with the protocol of Section 3.

C Robustness of the Voting Result

Table 2: Calibrated slope for Proposition 2 at $k = 8$ under alternative conditions. †: within the null range.

Condition	Qwen 0.5B	Qwen 1.5B	Qwen 7B	Qwen 14B	Llama 8B	Mistral 7B
Math-Verify (main)	0.86	0.80	0.70	0.85	0.84	0.82
exact-match scorer	0.90	0.80	0.65	0.76	0.97	0.84
MATH only	0.92	0.77	0.59	0.72	0.97	0.92
GSM8K only	0.75	0.77	0.12†	-0.24†	0.59	0.57
truncated removed	0.79	0.82	0.61	0.71	0.82	0.81

D Reward Model and Generator Scale at Matched Difficulty

Table 3: Difference in the reward model's AUC and best-of-8 accuracy between two generators, on problems where both have correct and incorrect samples. Without matching, the 0.5B model's AUC is higher; with matching, it is lower.

Pair	n	ΔAUC	Δbest-of-8
Qwen 0.5B − Qwen 14B	146	-0.092 [-0.139, -0.043]	-0.229 [-0.295, -0.162]
Qwen 1.5B − Qwen 14B	190	-0.029 [-0.062, +0.004]	-0.122 [-0.171, -0.077]
Qwen 7B − Qwen 14B	199	+0.016 [-0.007, +0.038]	-0.003 [-0.038, +0.035]

E Exchangeability of the Two Halves

Table 4: Difference in unbiased $\mathrm{pass@}k$ between the two halves of each problem's samples (second minus first, points).

Model	$k = 1$	$k = 2$	$k = 4$	$k = 8$	$k = 16$
Qwen 0.5B	-0.7 [-1.3, -0.0]	-0.9 [-1.6, -0.2]	-1.1 [-2.0, -0.2]	-1.5 [-2.7, -0.2]	-1.9 [-3.7, -0.1]
Qwen 1.5B	+0.3 [-0.3, +0.8]	+0.1 [-0.5, +0.7]	+0.1 [-0.6, +0.8]	+0.2 [-0.7, +1.1]	+0.7 [-0.6, +2.0]
Qwen 7B	-0.2 [-0.6, +0.2]	-0.3 [-0.7, +0.2]	-0.4 [-0.9, +0.1]	-0.5 [-1.1, +0.1]	-0.5 [-1.4, +0.4]
Qwen 14B	-0.1 [-0.4, +0.3]	-0.2 [-0.6, +0.1]	-0.4 [-0.8, +0.1]	-0.5 [-1.0, +0.0]	-0.6 [-1.3, +0.1]
Llama 8B	+0.6 [-0.0, +1.2]	+0.4 [-0.3, +1.0]	+0.2 [-0.6, +1.0]	+0.0 [-1.0, +0.9]	-0.4 [-1.6, +1.0]
Mistral 7B	+0.6 [-0.0, +1.2]	+0.8 [+0.0, +1.6]	+0.8 [-0.3, +1.9]	+0.3 [-1.1, +1.8]	-0.4 [-2.4, +1.7]

F Prediction Procedure and Its Validation

Accuracy is modeled as $p \sim \mathrm{Beta}(\alpha, \beta)$ with (α, β) fitted by maximum likelihood to the correct counts of all problems, and each problem's posterior is used for prediction, which corrects for selecting problems by their counts. Wrong answers observed at least twice are resampled in proportion to their counts; answers observed once are treated as new on every draw. Reward-model scores are transformed to the logit scale and modeled as normal with standard deviations pooled across problems with $n_i - 1$ degrees of freedom; the difference between the correct and incorrect means is shrunk

toward its population mean with a normal prior whose variance is estimated by the method of moments. On synthetic data drawn from a known model, the procedure predicted coverage, voting and best-of-k accuracy at $k \leq 128$ to within about one point. Two simpler versions, resampling observed scores and omitting the shrinkage, underpredicted best-of-k accuracy by up to seven points, because they cannot produce scores beyond the largest one observed.

G Process Reward Model Aggregations

Table 5: Process reward model aggregated over steps by minimum, mean, last step and product: AUC / best-of-8 accuracy with one correct candidate (%).

Model	min	mean	last	product
Qwen 0.5B	0.877 / 74	0.884 / 74	0.898 / 76	0.896 / 77
Qwen 1.5B	0.897 / 78	0.896 / 78	0.884 / 75	0.911 / 80
Qwen 7B	0.858 / 73	0.859 / 72	0.840 / 67	0.871 / 74
Qwen 14B	0.833 / 70	0.828 / 69	0.796 / 63	0.846 / 71
Llama 8B	–	–	–	–
Mistral 7B	0.794 / 57	0.818 / 58	0.838 / 61	0.840 / 62

H Reanalysis of the Released Samples of Brown et al.

We use the released samples for Llama-3-8B-Instruct and Llama-3-70B-Instruct on MATH (128 problems) and GSM8K (127 problems), 10,000 per problem, generated with few-shot prompts, together with the released correctness labels. Correct samples form one answer class. For incorrect samples we extract the final answer (the text after "The final answer is" or the last boxed expression for MATH, the number after #### for GSM8K) and group them after the normalization of Appendix A; incorrect samples without an extractable answer (at most 6.3% per setting) do not vote. Each problem's samples are split into two halves of 5,000 at random. From the first half we compute the share of problems whose correct answer is more frequent than every wrong one (the limit of Proposition 2) and, for finite k, voting accuracy under multinomial draws from the empirical distribution; on the second half we measure voting accuracy by draws without replacement. For the circularity analysis, pools of n samples are drawn without replacement, ten times per problem; strata and outcomes follow Section 3. We fixed in advance that only settings with at least 30 problems in both hardest strata at every n enter the summary, which excludes both GSM8K settings.

Table 6: Majority voting on the released samples of Brown et al. (2024): predicted from 5,000 samples per problem / observed on the other 5,000 (%). The last row is the share of problems whose correct answer is more frequent than every wrong answer.

k	MATH, Llama-3-8B	MATH, Llama-3-70B	GSM8K, Llama-3-8B	GSM8K, Llama-3-70B
1	26.7 / 26.5	48.0 / 48.0	76.7 / 76.9	93.2 / 93.1
4	32.0 / 32.2	53.8 / 53.7	82.8 / 83.0	95.4 / 95.6
16	39.2 / 39.5	58.6 / 58.4	86.1 / 85.9	96.7 / 96.8
64	42.8 / 42.6	60.6 / 60.2	87.0 / 87.1	96.8 / 96.9
256	43.7 / 43.7	61.0 / 60.7	87.8 / 87.7	96.9 / 96.9
1024	43.8 / 43.8	60.7 / 60.7	88.0 / 88.2	96.9 / 96.9
4096	43.8 / 43.8	60.4 / 60.0	88.2 / 88.8	96.9 / 96.9
∞ (Prop. 2)	43.8	60.9	88.2	96.9

I A Negative Result: Letting the Verifier Vote Against Answers

If scores are independent given correctness and calibrated, the Bayes-optimal way to combine them is to add each sample's log-likelihood ratio to its answer's total, so that low scores count against an answer (Nitzan & Paroush, 1982); unlike majority or reward-weighted voting, this rule cannot be captured by a frequent wrong answer. We calibrated the ratio on held-out problems and tested it

Table 7: Voting minus random choice at $k = 8$ on the hardest stratum ($p \leq 1/8$), circular / held-out (points), for pools of n samples drawn from the released samples; number of problems in the circular stratum in parentheses. GSM8K has too few such problems for a reliable estimate.

n	MATH, Llama-3-8B	MATH, Llama-3-70B	GSM8K, Llama-3-8B	GSM8K, Llama-3-70B
16	-1.3 / +8.2 (69)	-1.2 / +3.1 (49)	-4.0 / +6.0 (18)	-12.5 / -8.3 (1)
32	+1.1 / +6.3 (72)	-0.9 / +4.0 (49)	-3.6 / +0.8 (13)	-12.5 / -21.2 (1)
64	+0.2 / +5.0 (64)	+0.5 / +2.6 (46)	-2.7 / -2.3 (14)	-1.6 / -8.3 (1)
128	+0.5 / +1.9 (66)	+0.3 / +2.3 (45)	-1.7 / -2.5 (10)	–
256	+0.5 / +1.6 (67)	+1.0 / +2.1 (43)	+1.0 / -0.3 (7)	-0.4 / +0.0 (1)
512	+0.5 / +1.0 (67)	+0.4 / +1.2 (42)	-2.1 / +1.8 (6)	-0.2 / +0.0 (1)
1024	+0.3 / +1.0 (63)	+0.9 / +1.7 (40)	+0.5 / -2.4 (6)	-0.1 / -0.1 (1)

before looking at the results, with the preregistered criterion that it beat both voting and best-of-k at $k = 128$ on the fresh samples of Section 4.4. It did not: 65.3% against 78.5% for best-of-k (Table 8). On synthetic data the rule fails in the same way when scores shift by problem, which pushes every sample of a problem to the same side of the calibrated threshold and favors answers seen once. Reward-weighted voting also fell below best-of-k. Aggregating verifier evidence in a way that is robust to such shifts remains open.

Table 8: Aggregating verifier scores across samples (Qwen2.5-7B, 128 fresh samples, reward-model scores): accuracy (%) of majority voting, best-of-k, reward-weighted voting and likelihood-ratio voting, with coverage for reference.

k	Voting	Best-of-k	Weighted	Likelihood ratio	Coverage
4	49.9	61.8	61.3	59.0	75.2
8	56.1	68.4	67.6	62.1	85.8
16	59.9	73.1	71.1	63.8	93.3
32	62.4	75.6	72.9	64.7	97.6
64	64.1	77.7	73.9	65.7	99.3
128	64.6	78.5	74.3	65.3	100.0

J Additional Results

Figure 3 and Tables 9–15 give the full results summarized in the main text, generated from the same logged runs.

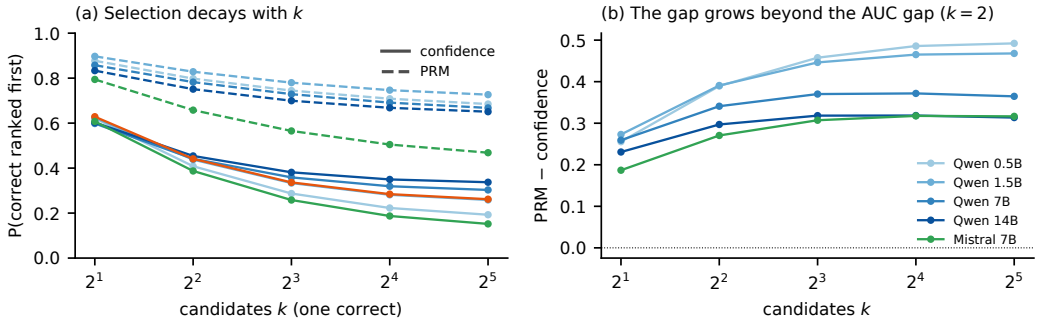


Figure 3: **(a)** Probability that a single correct candidate is ranked above all $k - 1$ incorrect ones, for model confidence (solid) and the process reward model (dashed); $k = 2$ is the AUC. **(b)** The difference between the two verifiers, which grows well beyond its value at $k = 2$.

Table 9: Single-sample accuracy (%) by model and task, with the share of samples that reach the 2,048-token limit in parentheses (%).

Model	MATH 1	MATH 2	MATH 3	MATH 4	MATH 5	GSM8K s	GSM8K l
Qwen 0.5B	64 (0.8)	47 (1.9)	36 (2.8)	24 (3.2)	10 (4.5)	53 (0.3)	29 (0.8)
Qwen 1.5B	84 (0.2)	73 (0.8)	61 (1.9)	50 (1.6)	26 (3.0)	78 (0.0)	61 (0.4)
Qwen 7B	95 (0.0)	90 (0.2)	84 (0.6)	73 (0.2)	56 (1.1)	92 (0.0)	88 (0.0)
Qwen 14B	96 (0.0)	93 (0.0)	89 (0.3)	80 (0.1)	62 (0.8)	94 (0.0)	94 (0.0)
Llama 8B	81 (1.0)	67 (2.5)	56 (6.2)	42 (8.4)	21 (15.5)	86 (0.1)	77 (0.6)
Mistral 7B	41 (0.2)	25 (0.5)	14 (0.9)	10 (1.1)	3 (1.6)	58 (0.0)	36 (0.1)

Table 10: Best-of-8 selection accuracy (%) with m correct candidates, with the interval of Proposition 3 computed per problem and averaged. n: problems with both correct and incorrect samples.

Model	Verifier	n	AUC	$m = 1$	$m = 2$	$m = 3$
Qwen 0.5B	confidence (mean)	1061	0.622	29 [13, 62]	42 [16, 82]	51 [19, 90]
Qwen 0.5B	confidence (min)	1061	0.580	25 [9, 58]	38 [11, 79]	48 [14, 88]
Qwen 0.5B	probe	1061	0.566	24 [10, 57]	35 [12, 77]	44 [15, 86]
Qwen 0.5B	PRM (min)	547	0.877	74 [63, 88]	83 [66, 95]	86 [68, 97]
Qwen 1.5B	confidence (mean)	986	0.624	33 [14, 62]	48 [16, 82]	57 [20, 90]
Qwen 1.5B	confidence (min)	986	0.602	31 [10, 60]	46 [13, 81]	56 [16, 90]
Qwen 1.5B	probe	986	0.555	27 [10, 56]	40 [12, 76]	49 [15, 85]
Qwen 1.5B	PRM (min)	515	0.897	78 [68, 90]	86 [70, 96]	89 [72, 98]
Qwen 7B	confidence (mean)	563	0.598	36 [15, 60]	50 [17, 78]	58 [20, 87]
Qwen 7B	confidence (min)	563	0.575	32 [10, 57]	47 [12, 78]	57 [14, 87]
Qwen 7B	probe	563	0.546	31 [11, 55]	44 [13, 74]	52 [16, 83]
Qwen 7B	PRM (min)	274	0.858	73 [59, 86]	82 [61, 94]	86 [64, 96]
Qwen 14B	confidence (mean)	452	0.603	38 [16, 60]	52 [18, 79]	61 [21, 87]
Qwen 14B	confidence (min)	452	0.579	35 [11, 58]	50 [13, 78]	60 [16, 87]
Qwen 14B	probe	452	0.533	31 [9, 53]	44 [11, 73]	54 [14, 83]
Qwen 14B	PRM (min)	230	0.833	70 [55, 83]	80 [57, 93]	84 [60, 96]
Llama 8B	confidence (mean)	987	0.629	34 [15, 63]	47 [17, 82]	57 [21, 90]
Llama 8B	confidence (min)	987	0.623	33 [13, 62]	48 [15, 82]	58 [19, 90]
Llama 8B	probe	987	0.602	32 [14, 60]	45 [16, 79]	54 [19, 88]
Mistral 7B	confidence (mean)	1011	0.607	26 [12, 61]	38 [14, 80]	48 [17, 89]
Mistral 7B	confidence (min)	1011	0.573	23 [8, 57]	35 [10, 78]	45 [13, 88]
Mistral 7B	probe	1011	0.617	26 [13, 62]	38 [15, 81]	47 [19, 89]
Mistral 7B	PRM (min)	260	0.794	57 [45, 79]	67 [48, 91]	73 [51, 94]

Table 11: Share (%) of the correct answers added by doubling the budget from k to $2k$ on hard problems (held-out stratum with 1–5 correct of 16) that the selector does not return, with 95% intervals. Doubling from 8 to 16 uses all of the held-out half and has a small denominator.

	Voting			PRM		
Model	2→4	4→8	8→16	2→4	4→8	8→16
Qwen 0.5B	76 [73, 80]	61 [55, 66]	68 [58, 77]	35 [29, 41]	38 [31, 45]	36 [24, 47]
Qwen 1.5B	79 [75, 84]	67 [59, 74]	61 [44, 74]	42 [35, 50]	41 [30, 50]	44 [23, 61]
Qwen 7B	88 [81, 96]	84 [74, 92]	86 [68, 102]	55 [42, 68]	59 [43, 74]	74 [49, 101]
Qwen 14B	92 [83, 100]	83 [71, 94]	77 [47, 100]	71 [55, 85]	66 [48, 83]	21 [-42, 71]
Llama 8B	74 [70, 78]	53 [45, 61]	38 [18, 54]	–	–	–
Mistral 7B	72 [69, 75]	50 [45, 56]	41 [28, 52]	50 [41, 60]	54 [43, 65]	59 [42, 73]

Table 12: Voting minus random choice at $k = 8$ (points) by difficulty: circular protocol / held-out protocol.

Model	$p \leq 1/8$	$1/8 < p \leq 5/16$	$5/16 < p \leq 5/8$
Qwen 0.5B	-0.8 / +7.1	+20.0 / +19.4	+35.5 / +28.4
Qwen 1.5B	-0.8 / +4.9	+14.9 / +17.1	+34.8 / +26.5
Qwen 7B	-2.3 / +1.3	+4.2 / +11.4	+26.7 / +21.0
Qwen 14B	-3.6 / +0.4	+6.6 / +9.7	+23.4 / +18.8
Llama 8B	+0.0 / +8.7	+20.3 / +20.9	+34.6 / +27.8
Mistral 7B	+2.0 / +10.1	+23.2 / +21.9	+39.9 / +32.3

Table 13: Held-out test of Proposition 2. For each budget: observed slope / null mean [null 95% range] from 30 null replicates. Last column: z and Holm-corrected p at $k = 8$ from 200 null replicates (the smallest attainable uncorrected p is $1/201$).

Model	$k = 4$	$k = 8$	$k = 16$	z / p_{Holm}
Qwen 0.5B	0.50 / -0.05 [-0.19, +0.11]	0.77 / -0.10 [-0.30, +0.21]	0.84 / -0.10 [-0.34, +0.08]	7.5 / 0.030
Qwen 1.5B	0.61 / +0.00 [-0.14, +0.17]	0.82 / -0.01 [-0.18, +0.22]	0.83 / -0.01 [-0.32, +0.32]	7.3 / 0.030
Qwen 7B	0.66 / +0.09 [-0.12, +0.28]	0.85 / +0.16 [-0.17, +0.37]	0.76 / +0.20 [-0.14, +0.61]	3.7 / 0.030
Qwen 14B	0.72 / +0.03 [-0.14, +0.24]	0.88 / -0.01 [-0.29, +0.30]	0.82 / +0.05 [-0.38, +0.43]	4.7 / 0.030
Llama 8B	0.58 / -0.02 [-0.18, +0.14]	0.85 / -0.08 [-0.37, +0.15]	0.81 / -0.09 [-0.38, +0.25]	7.0 / 0.030
Mistral 7B	0.35 / -0.09 [-0.25, +0.09]	0.67 / -0.13 [-0.41, +0.16]	0.70 / -0.22 [-0.55, +0.10]	7.1 / 0.030

Table 14: Prediction from 32 samples versus observation on 128 fresh samples (144 problems): predicted / observed accuracy (%).

k	Coverage	Voting	PRM
1	40.8 / 40.9	41.0 / 40.9	41.0 / 40.9
2	59.1 / 59.2	40.9 / 40.8	52.6 / 52.0
4	74.9 / 75.0	48.9 / 50.0	61.9 / 61.7
8	85.8 / 86.1	55.0 / 56.0	68.5 / 68.5
16	92.7 / 93.3	58.2 / 59.8	73.1 / 72.9
32	96.7 / 97.4	59.5 / 62.5	76.8 / 75.7
64	98.7 / 99.3	60.1 / 63.9	79.0 / 77.3
128	99.5 / 100.0	60.8 / 65.1	81.1 / 78.5

Table 15: Probability (%) that the selector returns a correct sample from eight held-out samples, given that one is present, by the number of correct samples in the other half.

Model	Stratum	n	Random	Voting	Confidence	Probe
Qwen 0.5B	1-2	247	20	29	23	23
Qwen 0.5B	3-5	187	30	54	38	34
Qwen 0.5B	6-10	226	50	81	62	55
Qwen 0.5B	11-15	282	80	98	88	83
Qwen 1.5B	1-2	127	21	29	22	24
Qwen 1.5B	3-5	115	31	51	40	32
Qwen 1.5B	6-10	216	53	83	63	56
Qwen 1.5B	11-15	399	83	99	89	84
Qwen 7B	1-2	52	20	20	25	22
Qwen 7B	3-5	53	32	43	36	32
Qwen 7B	6-10	98	54	78	55	57
Qwen 7B	11-15	265	86	99	88	85
Qwen 14B	1-2	36	22	22	25	23
Qwen 14B	3-5	38	30	42	33	29
Qwen 14B	6-10	68	52	74	59	52
Qwen 14B	11-15	232	88	99	92	87
Llama 8B	1-2	130	21	34	25	27
Llama 8B	3-5	126	31	55	38	37
Llama 8B	6-10	201	52	83	61	59
Llama 8B	11-15	404	84	99	90	88
Mistral 7B	1-2	280	20	37	27	29
Mistral 7B	3-5	191	31	58	41	42
Mistral 7B	6-10	235	50	86	61	62
Mistral 7B	11-15	182	77	98	87	83