INFORMATION-RATE LIMITS OF TEST-TIME REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time reasoning systems spend compute by sampling candidate solutions and using a verifier to select among them, but more samples help only when the selection signal can identify the correct rollout. We formalize this coverage–precision gap by modeling verifier-guided inference as adaptive hypothesis testing through a feedback channel. The verifier's per-query information capacity yields an algorithm-independent lower bound on the number of queries needed for reliable recovery, with matching behavior in query-rich settings and collapse when the verifier is uninformative. Empirically, we hold rollouts fixed and vary only the selector across math, code, and multiple-choice benchmarks and two model families: oracle selectors expose latent gains, while practical selectors recover only part of them and corrupted verifiers collapse toward the single-sample baseline. A corrupted-oracle calibration places practical selectors – a trained process reward model, majority voting, an LLM-as-judge verifier, and log-probability ranking – on a common capacity-equivalent axis. Where information is directly measurable, the trained PRM and judge order consistently by realized information and captured gain, turning verifier quality into a measurable bottleneck for test-time compute.

1 INTRODUCTION

Inference-time scaling has become a central phenomenon in modern language-model reasoning (Snell et al., 2024; Brown et al., 2024; Wu et al., 2025). Systems improve accuracy by sampling many candidate answers, voting across independent chains, running tree search over partial solutions, executing tools, or using learned reward models to select among rollouts. The empirical message is clear: more test-time compute can help. The basic theoretical question is harder: how much can it help, and what controls the point at which it stops?

Brown et al. (2024) make this question concrete. When solving MATH with Llama-3-8B-Instruct, the fraction of problems for which at least one of N sampled rollouts is correct (their notion of *coverage*) rises from 82.9% at $N = 100$ to 98.4% at $N = 10{,}000$. Yet the same study shows that the realised accuracy under common verifiers – majority voting and learned reward models – moves from 40.5% to only 41.4% over the same budget. Coverage scales over four orders of magnitude; the verifier's ability to find the correct sample plateaus after a hundred or so. This coverage–precision gap is the empirical phenomenon we want to understand and predict.

Existing analyses approach this question from two angles. Empirical scaling laws fit best-of-N or tree-search curves (Snell et al., 2024; Brown et al., 2024; Wu et al., 2025), but they describe specific samplers rather than fundamental limits; Wu et al. (2025) explicitly note that voting-based strategies converge *exponentially* to a fixed accuracy ceiling determined by the model's output distribution, but do not separate this ceiling from a property of the verifier. Classical sequential analysis (Naghshvar & Javidi, 2013; Jamieson & Nowak, 2011; Karp & Kleinberg, 2007) provides adaptive lower bounds for noisy hypothesis testing, but it has not been connected to the verifier signal that drives modern LLM inference. This leaves three concrete gaps for a predictive theory of test-time compute. *(i) Algorithm independence:* existing limits are specific to a sampler or selector, while real systems mix best-of-N, tree search, tools, and judge models. *(ii) Verifier heterogeneity:* oracle labels, reward models, public tests, process verifiers, and self-consistency convey very different amounts of information, but no common quantity captures this. *(iii) Saturation:* we lack a principled explanation of when more

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.

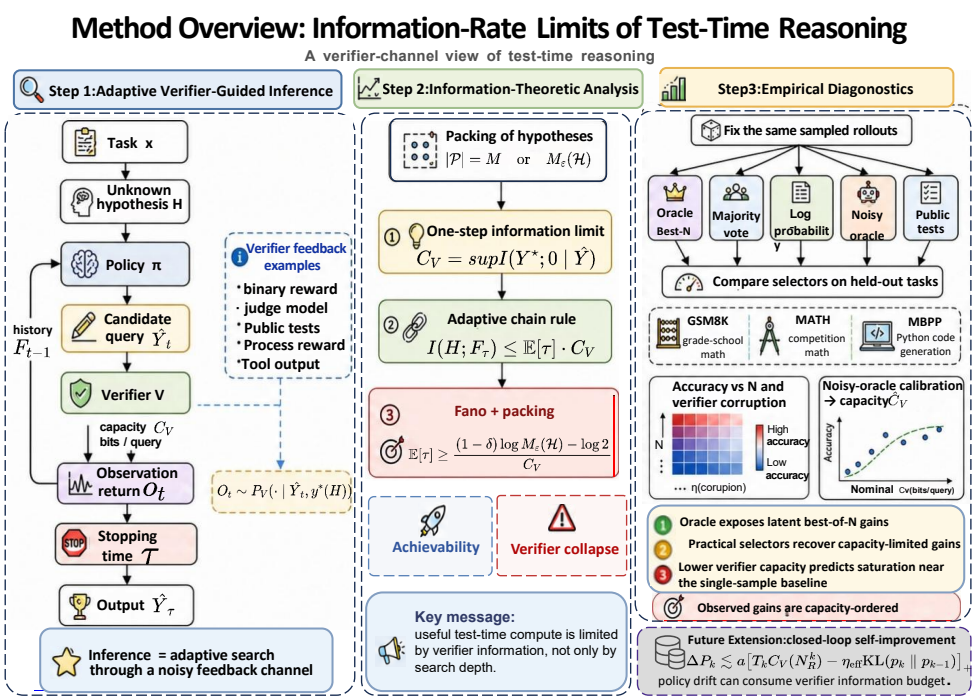

Figure 1: **Framework overview.** We view test-time reasoning as an adaptive verifier-query loop, analyze its per-query information limit through C_V, and evaluate the resulting capacity bottleneck with fixed-rollout selector diagnostics.

compute must stop helping and when it can keep paying off. Equivalently, no existing theory predicts *how many bits of feedback per query* translate into how much realisable best-of-N accuracy.

This paper addresses these gaps by treating verifier-guided inference as adaptive hypothesis testing through a feedback channel. The unknown solution is a hypothesis $H \in \mathcal{H}$. At time t, an inference policy chooses a candidate $\hat{Y}_t$, queries a verifier kernel V, observes $O_t \sim P_V(\cdot \mid \hat{Y}_t, y^\star(H))$, and either continues or stops at a stopping time τ. The verifier's one-step capacity $C_V = \sup_{P_{\hat{Y}, Y^\star}} I(Y^\star; O \mid \hat{Y})$ upper-bounds the useful information any single query can reveal about the true answer. Combining this capacity bound with Fano's inequality yields a lower bound of order $\log M / C_V$ on the number of inference steps needed to solve a packing of size M. The lens is uniform: best-of-N, self-consistency, tree search, code agents with public tests, and ORM/PRM-guided search (Cobbe et al., 2021; Uesato et al., 2022; Lightman et al., 2024) all become instances of the same channel-limited search problem, and the coverage–precision gap of Brown et al. (2024) becomes the simplest empirical witness of C_V being smaller than 1 bit.

Contributions.

- **Adaptive-query model.** We model verifier-guided inference as adaptive hypothesis testing with a verifier kernel and a stopping time, and we identify the per-query verifier capacity C_V as the controlling quantity. This abstraction unifies best-of-N, self-consistency, tree search, code agents, and verifier-guided refinement.
- **Verifier-capacity lower bound.** We prove that every adaptive policy satisfies $\mathbb{E}[\tau] \ge ((1-\delta)\log M - \log 2)/C_V$, including policies with data-dependent stopping and history-dependent candidate generation (Theorem 1).
- **Achievability and collapse.** We give a posterior-splitting achievability result for explicitly information-rich verifiers (Proposition 1) and a verifier-collapse impossibility (Theorem 2), delineating when the lower bound is tight and when more compute cannot help.

- **Empirical diagnostics.** Holding rollouts fixed and varying only the selector on GSM8K, MATH, MBPP, and 13 multiple-choice datasets, realized best-of-N gains track the information available to the selector, and accuracy collapses back toward pass@1 as controlled verifier capacity decreases. The ordering replicates on a second family (Llama-3.1-8B, including a low-pass@1 MATH setting with $+29.6$ points of latent gain) and across Qwen2.5 from 1.5B to 14B. We further turn the noisy-oracle curve into a calibration tool assigning capacity-equivalent scores to practical selectors on the fixed GSM8K reference curve: about 0.30 for a trained process reward model (Qwen2.5-Math-PRM-7B), 0.21 for majority vote, 0.10 for a Qwen2.5 LLM judge, and 0.03 for log-probability (Sec. 6).

- **Operationalized packing size.** We estimate an effective per-problem packing size $\hat{M}_{\text{eff}}$ from the answer distribution of the fixed rollouts and show that accuracy loss under capacity-limited selection concentrates where $\log \hat{M}_{\text{eff}}/C_V$ is large, connecting the rate in Theorem 1 to measured accuracy curves (Sec. 6).

The empirical contribution is a diagnostic, not a leaderboard: every main experiment fixes the sampled candidates and changes only the channel used to select among them, isolating the mechanism that sampling determines *coverage* while verifier information determines how much coverage becomes realized accuracy.

What is new and what is standard. The mathematical machinery – Fano, data processing, and a per-step information budget for adaptive queries – is the standard active-sequential-testing toolkit (Naghshvar & Javidi, 2013; Jamieson & Nowak, 2011); Lemma 1 specializes it to verifier kernels with data-dependent stopping, and Theorem 2 is a two-point argument. The contribution is the abstraction and its consequences: the verifier as a feedback channel whose one-step capacity C_V controls LLM test-time compute, the algorithm-independent reading of the coverage–precision gap, and the fixed-rollout diagnostic that makes C_V comparable to practical selectors on a stated calibration curve.

Notation. Unless stated otherwise, logarithms are base two, so mutual information and capacity are measured in bits.

2 Related Work

Inference-time compute has become a first-class scaling axis: best-of-N, self-consistency, tree search, and prompt-adaptive allocation can substantially improve reasoning accuracy (Snell et al., 2024; Brown et al., 2024; Wu et al., 2025; Yao et al., 2023; Wang et al., 2023; 2025; Muennighoff et al., 2025). Existing scaling laws and allocation methods are mostly sampler-specific; our goal is complementary: we ask for an algorithm-independent rate at which verifier-guided inference can convert compute into reliable progress.

Modern reasoning systems also scale verification: outcome reward models (Cobbe et al., 2021), process reward models (Uesato et al., 2022; Lightman et al., 2024; Setlur et al., 2024), public unit tests, tools, and LLM judges all provide feedback over sampled candidates. Recent systems explicitly improve verification alongside generation (DeepSeek-AI, 2025; Zhao et al., 2025). We model these mechanisms as observation channels $P_V(O \mid \hat{y}, y^\star)$ and identify their effective information rate C_V as the quantity that controls compute scaling.

Our analysis builds on classical information-theoretic tools: Fano's inequality, data processing, Yao's minimax principle, active sequential testing, posterior splitting, and noisy-oracle search (Cover & Thomas, 2006; Tsybakov, 2009; Polyanskiy & Wu, 2024; Yao, 1977; Naghshvar & Javidi, 2013; Jamieson & Nowak, 2011; Singer & Vondrák, 2015; Karp & Kleinberg, 2007); the demarcation from active sequential testing (Naghshvar & Javidi, 2013) is stated above. Our setting also differs from best-of-N alignment analyses (Beirami et al., 2024; Amini et al., 2024; Yang et al., 2024), which study distributional effects under fixed rewards; here the feedback channel itself is the bottleneck.

3 ADAPTIVE VERIFIER-GUIDED INFERENCE

3.1 TASKS, HYPOTHESES, AND VERIFIER FEEDBACK

Let $x \sim \mathcal{D}$ be a task instance and let $y^\star \in \mathcal{Y}$ be its correct answer. We model the uncertainty relevant to inference by a hypothesis class $\mathcal{H}$. Each $h \in \mathcal{H}$ determines a target answer $y^\star(h)$. The policy does not observe h directly.

Definition 1 (Verifier kernel)**.** *A verifier is a Markov kernel*

$$V : \mathcal{Y} \times \mathcal{Y} \to \Delta(\mathcal{O}), \qquad O \mid (\hat{y}, y^\star) \sim P_V(\cdot \mid \hat{y}, y^\star),$$

where $\hat{y}$ is the candidate query and O is the feedback returned to the policy.

This definition includes binary outcome rewards, noisy reward models, process rewards, public unit tests, tool observations, and degenerate self-consistency signals. For a binary correctness verifier with crossover probability $\eta < 1/2$, the channel capacity is $1 - h_2(\eta)$ bits. For an uninformative self-consistency signal on novel tasks, the realized information rate can be zero.

Definition 2 (Inference policy)**.** *At step t, a policy π observes the history*

$$\mathcal{F}_{t-1} = \sigma(\hat{Y}_1, O_1, \ldots, \hat{Y}_{t-1}, O_{t-1}, U),$$

where U is private randomness. It chooses a candidate $\hat{Y}_t$, receives $O_t \sim P_V(\cdot \mid \hat{Y}_t, y^\star(H))$, and decides whether to stop. The stopping time is τ, and the final output is $\hat{Y}_\tau$.

The policy succeeds at error δ over a finite packing $\mathcal{P} \subseteq \mathcal{H}$ if

$$\sup_{h \in \mathcal{P}} \mathbb{P}_h[\hat{Y}_\tau \neq y^\star(h)] \leq \delta.$$

Its test-time cost is $\mathbb{E}[\tau]$.

3.2 VERIFIER CAPACITY

Definition 3 (One-step verifier capacity)**.** *The capacity of a verifier is*

$$C_V := \sup_{P_{\hat{Y}, Y^\star}} I(Y^\star; O \mid \hat{Y}),$$

where the supremum ranges over joint laws of $(\hat{Y}, Y^\star)$ and $O \sim P_V(\cdot \mid \hat{Y}, Y^\star)$.

This is an upper bound on how many bits about the hidden answer any one verifier call can transmit after the candidate query is known. The conditioning on $\hat{Y}$ is important: the policy may choose queries adaptively, but the information gained from the verifier response itself is still bounded by C_V.

Capacity versus realized information. The quantity C_V is a worst-case channel capacity. It is useful for lower bounds because it upper-bounds every possible adaptive policy, but it may overestimate the information realized by a particular model, prompt, and task distribution. A sharper but policy-dependent quantity is

$$\bar{C}_V(\pi, \mathcal{D}, T) := \frac{1}{T} \sum_{t=1}^{T} I(Y^\star; O_t \mid \hat{Y}_t, \mathcal{F}_{t-1}),$$

the average information actually obtained along the rollout distribution of π. Our formal theorems use C_V; our empirical section uses controlled capacity proxies, such as $1 - h_2(\eta)$ for a corrupted oracle, and reports selection gaps for practical selectors such as majority vote, log-probability, and public tests.

Three distinct information quantities. Because the empirical section reports several numbers "in bits," we fix the taxonomy here. *(i) Worst-case capacity* C_V is a supremum over input laws; it is the correct quantity for lower bounds but can exceed the information realized by any particular policy. *(ii) Realized mutual information* is $I(Y^\star; O)$ evaluated at the empirical joint distribution of candidates

and feedback; it depends on the (possibly heavily skewed) input prior. *(iii) Capacity-equivalent scores* C_{eq} (Sec. 6) are calibrated comparisons: the capacity of the corrupted oracle whose captured oracle gain matches the selector's on the same fixed rollouts. These three quantities need not coincide, and reporting them separately resolves apparent discrepancies. For the LLM judge in Sec. 6, the empirical confusion matrix (true-positive rate 0.91, false-positive rate 0.65) has channel capacity 0.078 bits – close to its capacity-equivalent score $C_{\mathrm{eq}} \approx 0.10$ – while its realized mutual information is 0.036 bits, smaller only because the empirical input distribution is dominated by already-correct candidates (89%), far from the capacity-achieving prior.

4 Main Lower Bound

The core technical issue is adaptivity. The policy can choose each candidate after observing all previous verifier outputs, and the stopping time itself may depend on the observations. Nevertheless, the total mutual information cannot grow faster than C_V per non-stopped step.

Lemma 1 (Adaptive chain rule)**.** *Let H be any random hypothesis independent of the policy's private randomness, and let π be any adaptive inference policy with finite-mean stopping time τ. Then*

$$I(H; \mathcal{F}_\tau) \le \mathbb{E}[\tau]\, C_V.$$

Proof sketch. Expose the policy's private randomness and pad the transcript with a null symbol after the stopping time. Applying the chain rule to this padded transcript, the candidate at round t contributes no fresh information about H: on the event $\{\tau \ge t\}$ it is measurable with respect to the past transcript and the private randomness, and on $\{\tau < t\}$ it is the null symbol. Thus the only possible increment is the verifier response. Conditional on any past history that has not stopped, the verifier response is drawn from the same kernel $P_V(\cdot \mid \hat{Y}_t, y^\star(H))$, so its conditional mutual information is at most C_V by definition. The tth increment is therefore bounded by $\mathbb{P}[\tau \ge t]C_V$. Summing over t gives $C_V \sum_t \mathbb{P}[\tau \ge t] = C_V \mathbb{E}[\tau]$, and data processing transfers the bound from the padded transcript to the stopped history. Appendix A gives the full stopping-time argument.

Theorem 1 (Verifier-capacity lower bound)**.** *Let $\mathcal{P} \subseteq \mathcal{H}$ have size M and pairwise distinct targets $\{y^\star(h) : h \in \mathcal{P}\}$. If an adaptive inference policy achieves error at most δ on every $h \in \mathcal{P}$, then*

$$\mathbb{E}[\tau] \ge \frac{(1-\delta)\log M - \log 2}{C_V}.$$

More generally, if success means ε-recovery in a metric answer space, M can be replaced by the ε-packing number $M_\varepsilon(\mathcal{H})$.

Proof sketch. Draw $H \sim \mathrm{Unif}(\mathcal{P})$; since the targets in $\mathcal{P}$ are distinct, recovering the answer identifies the packing element. Fano's inequality lower-bounds the error by $1 - (I(H;\hat{Y}_\tau) + \log 2)/\log M$, while data processing and Lemma 1 give $I(H;\hat{Y}_\tau) \le I(H;\mathcal{F}_\tau) \le \mathbb{E}[\tau]C_V$. Rearranging yields the claim; the metric case replaces $\mathcal{P}$ by an ε-packing. Appendix A gives the computation in full.

Corollary 1 (Noisy outcome verifier)**.** *For a binary-symmetric outcome verifier with crossover probability η,*

$$\mathbb{E}[\tau] \ge \frac{(1-\delta)\log M - \log 2}{1 - h_2(\eta)} = \Omega\!\left(\frac{\log M}{(1-2\eta)^2}\right) \quad \text{as } \eta \to 1/2.$$

This corollary makes the low-capacity-verifier penalty explicit. As the verifier approaches random guessing, the amount of test-time compute required for the same success target diverges.

The rate is not an artifact of the bound. It is matched from above in query-rich regimes. Call a verifier C-rich if, whenever the posterior has not concentrated, the policy can choose an admissible query whose next observation carries at least C bits about H – as in twenty-questions protocols, diagnostic test selection, and code tasks where unit tests split a version space. A standard posterior-splitting policy then achieves $\mathbb{E}[\tau] \le (\log M + \log(1/\delta))/C + O(\log\log M)$ (Proposition 1, Appendix A), so $\log M / C_V$ is the correct first-order rate whenever the verifier exposes sufficiently informative queries.

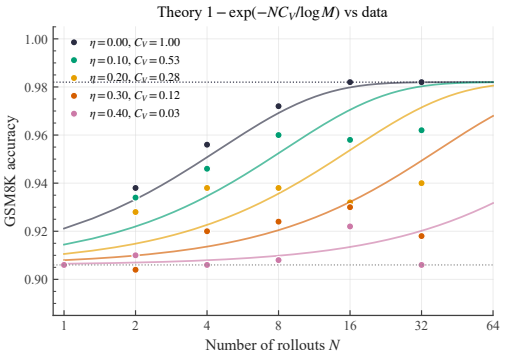

Figure 2: **Takeaway: higher verifier capacity gives a steeper scaling curve.** On the same GSM8K rollouts, noisy-oracle accuracies are ordered by $C_V \in \{1.00, 0.53, 0.28, 0.12, 0.03\}$ bits.

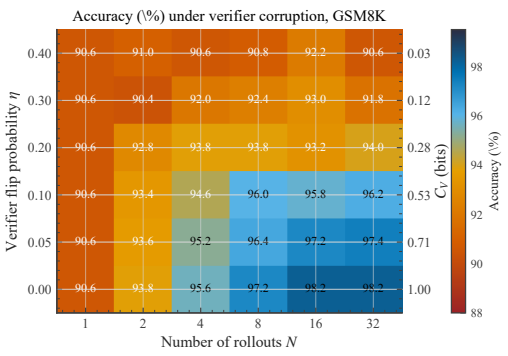

Figure 3: **Takeaway: rollout count helps only when the verifier remains informative.** On GSM8K, increasing N helps for a clean oracle but not for a strongly corrupted one.

5 Consequences for Test-Time Scaling

5.1 Verifier-conditioned saturation diagnostics

Let $p(N)$ denote the best possible success probability after expected compute N. Rewriting the lower bound implies that high accuracy is impossible until NC_V is comparable to the number of bits needed to identify the answer. In heterogeneous benchmarks, this produces a saturation curve: easy instances with small effective packing number are solved early, while hard instances require larger N or a stronger verifier.

The testable content is ordinal: the slope of realized test-time scaling should improve with verifier information, and the residual gap should reflect either limited selector information or irreducible task ambiguity. Figure 2 shows the ordering on noisy-oracle GSM8K data. Appendix A records the exponential functional form sometimes fitted to such curves and states precisely which extra assumptions it needs; we use it only as a diagnostic interpretation, never as a theorem.

5.2 When verifier information vanishes

Theorem 2 (Verifier collapse)**.** *Suppose there exist $h_0, h_1 \in \mathcal{H}$ with $y^\star(h_0) \neq y^\star(h_1)$ such that $P_V(\cdot \mid \hat{y}, y^\star(h_0)) = P_V(\cdot \mid \hat{y}, y^\star(h_1))$ for every query $\hat{y}$. Then no inference policy can distinguish h_0 from h_1 with error below $1/2$, regardless of compute.*

By induction the entire history then has identical law under h_0 and h_1, and Le Cam's two-point argument lower-bounds the error by $1/2$. This formalizes the zero-information regime: if the verifier cannot see the difference that matters, more rollouts merely generate more indistinguishable observations.

5.3 Outcome vs. process verification

In the present framework the outcome/process distinction is a statement about effective per-step capacity: a horizon-H task with only a final noisy outcome signal spreads at most constant information over H latent decisions, whereas a process verifier may transmit useful information at each step. This is an information-theoretic reading of why process supervision produces large gains on compositional reasoning, and Sec. 6 is consistent with this prediction with a trained PRM.

Table 1: GSM8K information-rate diagnostic with Qwen2.5-7B-Instruct on 500 problems. Entries are point estimates in percent. The oracle shows that additional rollouts contain latent correct solutions; practical selectors recover different fractions of this gain. As the oracle verifier is corrupted, accuracy moves back toward pass@1 in line with decreasing $C_V = 1 - h_2(\eta)$. "n/a" means that the practical selector has no prescribed channel capacity; its capacity-equivalent score is obtained from the calibration in Figure 4.

Selector	C_V (bits)	Acc.@1	Acc.@8	Acc.@32	Residual gap
Oracle best-of-N	1.000	90.6	97.2	98.2	0.0
PRM (Qwen2.5-Math-PRM-7B)	n/a	90.6	93.4	94.2	4.0
Majority vote	n/a	90.6	92.0	93.0	5.2
Log-prob selector	n/a	90.6	90.6	90.6	7.6
Noisy oracle, $\eta = 0.1$	0.531	90.6	96.0	96.2	2.0
Noisy oracle, $\eta = 0.2$	0.278	90.6	93.8	94.0	4.2
Noisy oracle, $\eta = 0.3$	0.119	90.6	92.4	91.8	6.4
Noisy oracle, $\eta = 0.4$	0.029	90.6	90.8	90.6	7.6

6 Empirical Diagnostic: Verifier Information Controls Best-of-N Gains

The theory predicts that rollouts help only when the selector can identify a correct candidate. We therefore use a fixed-rollout diagnostic: the candidate set is held fixed, and only the selector channel changes. On 500 GSM8K problems with 32 Qwen2.5-7B-Instruct completions each, we compare first sample, majority vote, normalized log-probability, oracle best-of-N, and noisy oracles with capacity $C_V(\eta) = 1 - h_2(\eta)$.

Table 1 shows the central pattern. The candidate set contains latent gains: pass@1 is 90.6%, while oracle best-of-32 reaches 98.2%. Practical selectors recover only part of this gap, and reducing verifier capacity moves accuracy monotonically back toward the single-sample baseline. Thus increasing N is not sufficient by itself; the gain depends on the information content of the verifier channel.

Capacity-equivalent scores for practical selectors. The corrupted-oracle results in Table 1 are an empirical calibration: they relate verifier capacity $C_V = 1 - h_2(\eta)$ to the captured fraction of the oracle gain at $N = 32$, on the same fixed rollouts. Reading this calibration backwards gives a *capacity-equivalent score*. Given any selector evaluated on the same fixed rollouts, we project its captured fraction onto the corrupted-oracle curve and report the capacity of the noisy oracle with matching captured gain. On GSM8K this gives $C_{\mathrm{eq}} \approx 0.30, 0.21, 0.10, 0.03$ for the trained PRM, majority vote, LLM judge, and log-probability (Figure 4). These are calibrated comparisons on one fixed-rollout reference curve, not selector-capacity or MI estimates.

To connect the calibration to learned verification, we evaluate a real LLM-as-judge on the same fixed rollouts: Qwen2.5-7B-Instruct scores each GSM8K candidate and the largest judged correctness probability is selected. The judge reaches 91.6% at $N = 32$ and captures 13.2% of the oracle gain. Its measured channel has capacity 0.078 bits, whereas realized MI is 0.036 bits (binary) or 0.046 bits (soft). Holding its TPR/FPR fixed, reweighting the skewed correctness prior from 0.890 to 0.5 raises MI to 0.078 bits; thus capacity, realized MI, and C_{eq} are distinct. On the less-skewed Llama MATH pool, MI is 0.161 bits against 0.175-bit capacity, although the channel also changes.

A trained process reward model. We further score the same fixed rollouts with Qwen2.5-Math-PRM-7B (Zhang et al., 2025), selecting the candidate with the highest mean step reward. The PRM reaches 94.2% at $N = 32$, capturing 47.4% of the oracle gain ($C_{\mathrm{eq}} \approx 0.30$; soft MI 0.12 bits), above the judge on both captured gain and MI. On Llama MATH (pass@1 62.8%), the same PRM reaches 82.8% at $N = 16$ and captures 67.6% of the 29.6-point oracle gain, versus 54.1% for majority, 50.0% for $\eta = 0.2$, and 31.1% for the judge. We omit C_{eq} there because the three-point MATH calibration is too coarse.

Cross-family replication. On Llama-3.1-8B-Instruct MATH, ARC-Challenge, and OpenBookQA, corrupted-oracle accuracy is monotone in capacity in every setting (Table 3). MATH supplies a

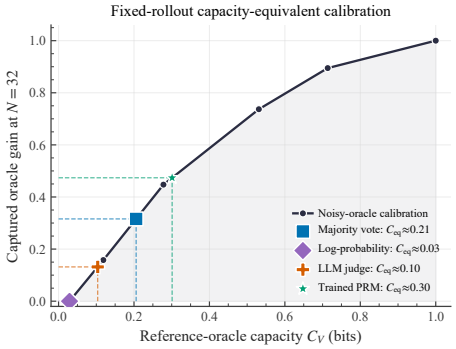

Figure 4: **Takeaway: practical selectors behave like low-capacity verifiers.** The noisy-oracle calibration maps a trained PRM, majority vote, a Qwen2.5 LLM judge, and log-probability to $C_{\text{eq}} \approx 0.30$, 0.21, 0.10, and 0.03 on the fixed GSM8K reference curve.

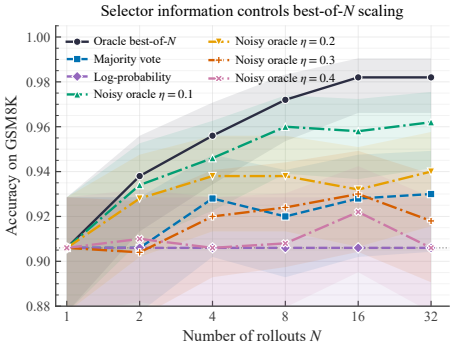

Figure 5: **Takeaway: the same rollouts give different gains depending on the selector.** Oracle best-of-N keeps recovering latent correct answers, while practical and corrupted selectors plateau below it.

low-pass@1 regime (62.8%; oracle 92.4%): $\eta = 0.2$ captures 50% of the latent gain versus 20% for $\eta = 0.4$.

Operationalizing the packing size. We estimate a per-problem effective packing size $\hat{M}_{\text{eff}} = 2^{H(\hat{p})}$ from the empirical answer distribution $\hat{p}$ of the candidates. Split-sample estimates avoid reusing candidates: on GSM8K, $\log_2 \hat{M}_{\text{eff}}$ correlates with held-out first-correct index at $\rho = 0.53$ $[0.45, 0.59]$ and capacity loss at 0.62 $[0.51, 0.72]$; on Llama MATH the corresponding values are 0.48 $[0.36, 0.59]$ and 0.46 $[0.34, 0.56]$. At fixed $\eta = 0.4$, loss is small for $\log_2 \hat{M}_{\text{eff}} \leq 0.5$ but large above 2 bits in both families (Table 4). Across eight model–dataset settings, correlations lie in $[0.48, 0.77]$; breadth-suite values $\hat{M}_{\text{eff}} = 1.06$–$1.45$ are consistent with its small headroom.

Scale. We repeat the diagnostic across Qwen2.5 at 1.5B, 3B, and 7B on four multiple-choice datasets (12 cells, ordering satisfied in 11 with one tie) and at 14B on MMLU, where $\eta = 0.2$ captures 57.1% of oracle gain versus 12.5% for $\eta = 0.4$. On the common 250-problem subset, headroom is not monotone in size (16.0, 14.8, 9.6, 22.4 points); it is better rank-aligned with $\hat{M}_{\text{eff}}$ (Spearman 0.8) than parameter count (0.2). We treat this four-point comparison as illustrative.

Statistical resolution. Paired 10,000-resample intervals on Llama MATH give PRM–majority +4.0 points $[0.4, 7.6]$ and $\eta = 0.2$–$\eta = 0.4$ +8.8 $[4.0, 14.0]$; majority–$\eta = 0.2$ includes zero. The corrupted-oracle curve is capacity-ordered in two rollout draws. Wide C_{eq} intervals and the coarse MATH grid motivate interpreting it only as a reference scale.

7 Discussion

More samples increase coverage; realized accuracy also requires verifier information on hard instances. Strengthening the verifier therefore targets the bottleneck after sampling saturates, although system-level cost efficiency remains task-dependent (Wu et al., 2025; Snell et al., 2024). Our results isolate fixed rollouts and verifiers; closed-loop policy updates and drift–capacity tradeoffs remain open.

Reproducibility and AI Assistance

Reproducibility. Complete proofs for all theoretical results are provided in Appendix A. Experimental setups, hyperparameters, dataset details, and evaluation procedures are described in Section 6 and Appendix B. All source code, seeds, and rollout evaluation artifacts are archived to support exact reproduction.

AI Assistance. Large language models were used solely for grammar, phrasing, and text polishing during the preparation of this manuscript. All conceptual formulations, theoretical proofs, experiment designs, and empirical analyses were conducted and verified by the authors.

REFERENCES

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

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. *arXiv preprint arXiv:2401.01879*, 2024.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020.

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.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2924–2936. Association for Computational Linguistics, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2006.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a.

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*, 2021b.

Kevin G. Jamieson and Robert Nowak. Active ranking using pairwise comparisons. In *Advances in Neural Information Processing Systems*, 2011.

Richard M. Karp and Robert Kleinberg. Noisy binary search and its applications. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 881–890, 2007.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. QASC: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8082–8090, 2020.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794. Association for Computational Linguistics, 2017.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison 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.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391. Association for Computational Linguistics, 2018.

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

Mohammad Naghshvar and Tara Javidi. Active sequential hypothesis testing. *The Annals of Statistics*, 41(6):2703–2738, 2013.

Yury Polyanskiy and Yihong Wu. *Information Theory: From Coding to Learning*. Cambridge University Press, 2024.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial Winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. SocialIQA: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 4463–4473. Association for Computational Linguistics, 2019.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for LLM reasoning. *arXiv preprint arXiv:2410.08146*, 2024.

Yaron Singer and Jan Vondrák. Information-theoretic lower bounds for convex optimization with erroneous oracles. In *Advances in Neural Information Processing Systems*, 2015.

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

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4149–4158. Association for Computational Linguistics, 2019.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Jian Wang, Boyan Zhu, Chak Tou Leong, Yongqi Li, and Wenjie Li. Scaling over scaling: Exploring test-time scaling plateau in large reasoning models. *arXiv preprint arXiv:2505.20522*, 2025.

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

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106. Association for Computational Linguistics, 2017.

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.

Joy Qiping Yang, Salman Salamatian, Ziteng Sun, Ananda Theertha Suresh, and Ahmad Beirami. Asymptotics of language model alignment. *arXiv preprint arXiv:2404.01730*, 2024.

Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science*, pp. 222–227, 1977.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800. Association for Computational Linguistics, 2019.

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.

Eric Zhao, Pranjal Awasthi, and Sreenivas Gollapudi. Sample, scrutinize and scale: Effective inference-time search by scaling verification. *arXiv preprint arXiv:2502.01839*, 2025.

A Proof Details for the Adaptive Lower Bound

We provide the full stopping-time argument for Lemma 1. Let U collect all private randomness of the policy, with $U \perp H$. For a deterministic horizon T, define a padded transcript by setting

$$\tilde{Y}_t = \begin{cases} \hat{Y}_t, & t \le \tau, \\ \perp, & t > \tau, \end{cases} \qquad \tilde{O}_t = \begin{cases} O_t, & t \le \tau, \\ \perp, & t > \tau. \end{cases}$$

Let

$$\tilde{\mathcal{F}}_t = \sigma(U, \tilde{Y}_1, \tilde{O}_1, \dots, \tilde{Y}_t, \tilde{O}_t).$$

Since τ is a stopping time, the event $\{\tau \ge t\}$ is measurable with respect to $\tilde{\mathcal{F}}_{t-1}$. Moreover, on this event the query $\hat{Y}_t$ is a measurable function of $(\tilde{\mathcal{F}}_{t-1}, U)$, while on $\{\tau < t\}$ the padded query is the deterministic null symbol. Hence

$$I(H; \tilde{Y}_t \mid \tilde{\mathcal{F}}_{t-1}) = 0$$

for each t. Applying the chain rule to the padded transcript gives

$$\begin{aligned} I(H; \tilde{\mathcal{F}}_T) &= \sum_{t=1}^{T} I(H; \tilde{Y}_t, \tilde{O}_t \mid \tilde{\mathcal{F}}_{t-1}) \\ &= \sum_{t=1}^{T} I(H; \tilde{O}_t \mid \tilde{\mathcal{F}}_{t-1}, \tilde{Y}_t). \end{aligned}$$

The tth summand decomposes according to whether the process has stopped:

$$I(H; \tilde{O}_t \mid \tilde{\mathcal{F}}_{t-1}, \tilde{Y}_t) = \mathbb{P}(\tau \ge t)\, I(H; O_t \mid \tilde{\mathcal{F}}_{t-1}, \hat{Y}_t, \tau \ge t),$$

because $\tilde{O}_t = \perp$ deterministically on $\{\tau < t\}$. Conditional on any past history with $\tau \ge t$, the verifier response follows the channel $P_V(\cdot \mid \hat{Y}_t, y^\star(H))$. Since the response depends on H only through $Y^\star = y^\star(H)$, and since C_V is the supremum of $I(Y^\star; O \mid \hat{Y})$ over all induced laws,

$$I(H; O_t \mid \tilde{\mathcal{F}}_{t-1}, \hat{Y}_t, \tau \ge t) \le C_V.$$

Therefore, for every finite T,

$$I(H; \tilde{\mathcal{F}}_T) \le C_V \sum_{t=1}^{T} \mathbb{P}(\tau \ge t).$$

Letting $T \to \infty$ and using monotone convergence together with $\sum_{t \ge 1} \mathbb{P}(\tau \ge t) = \mathbb{E}[\tau]$ yields

$$I(H; \tilde{\mathcal{F}}_\infty) \le \mathbb{E}[\tau] C_V.$$

Finally, the stopped history $\mathcal{F}_\tau$ is a measurable function of the padded infinite transcript, so data processing gives

$$I(H; \mathcal{F}_\tau) \le \mathbb{E}[\tau] C_V.$$

This proves Lemma 1.

A.1 Fano step for Theorem 1

It suffices to prove the statement under the uniform prior on the packing set: draw $H \sim \mathrm{Unif}(\mathcal{P})$ and run the policy on the corresponding instance. Because the targets in $\mathcal{P}$ are distinct, any successful target estimate also identifies the packing element, so the assumed worst-case error bound implies $\mathbb{P}[\hat{Y}_\tau \ne y^\star(H)] \le \delta$ under this prior. Fano's inequality gives

$$\mathbb{P}[\hat{Y}_\tau \ne y^\star(H)] \ge 1 - \frac{I(H; \hat{Y}_\tau) + \log 2}{\log M}.$$

The final answer is a function of the stopped transcript and the policy's private randomness, so data processing and Lemma 1 imply

$$I(H; \hat{Y}_\tau) \le I(H; \mathcal{F}_\tau) \le \mathbb{E}[\tau] C_V.$$

Combining the two displays with the upper bound δ on the error probability and rearranging yields

$$\mathbb{E}[\tau] \ge \frac{(1-\delta)\log M - \log 2}{C_V}.$$

For metric recovery, take an ε-packing of the hypothesis class. If the targets are pairwise separated at scale ε, then any ε-accurate output identifies the packing element, so the same Fano reduction applies with $M = M_\varepsilon(\mathcal{H})$.

Table 2: Cross-setting summary. Entries are point estimates in percent. "Oracle" is best-of-N on the same sampled rollouts. "Lightweight selector" is majority vote for math tasks and full public tests for MBPP. The gap between pass@1 and oracle is the latent test-time compute gain that a verifier channel could in principle extract.

Setting	N	Pass@1	Oracle	Lightweight selector	Oracle gain
GSM8K baseline	32	90.6	98.2	93.0	7.6
GSM8K replications	32	88.8	97.5	93.1	8.7
GSM8K high-diversity	32	88.3	98.7	93.0	10.3
MATH	16	83.2	93.2	90.4	10.0
MATH (Llama-3.1-8B)	16	62.8	92.4	78.8	29.6
MBPP code	16	69.0	81.5	80.0	12.5

A.2 Achievability of the rate

The lower bound is tight in query-rich regimes. Suppose that whenever the posterior has not concentrated, the policy can choose an admissible query whose next verifier observation carries at least C bits of mutual information about H; call such a verifier C-rich.

Proposition 1 (Posterior-splitting achievability)**.** *If $|\mathcal{H}| = M$ and the verifier is C-rich, then there is an adaptive policy such that*

$$\mathbb{E}[\tau] \le \frac{\log M + \log(1/\delta)}{C} + O(\log \log M).$$

The proof is the standard posterior-splitting argument: each query reduces posterior entropy by at least C bits until a single hypothesis has posterior mass $1 - \delta$. Thus the lower bound has the right first-order rate $\log M / C_V$ whenever the verifier exposes sufficiently informative queries.

A.3 The exponential saturation form, and what it assumes

Curves of the kind plotted in Figure 2 are often summarised by

$$1 - p(N) \gtrsim \exp\left(-\frac{N C_V}{\log M_\varepsilon(\mathcal{H})}\right) - \delta_0,$$

with δ_0 an irreducible Bayes-error term. We stress that this form does *not* follow from Fano alone: it additionally assumes a distribution of instance hardness, or an i.i.d. best-of-N success model. Our claims in the main text use only the ordinal consequence – higher verifier capacity yields a steeper realized scaling curve – which is what the experiments test.

B Additional Empirical Details

Uncertainty intervals in Sec. 6 use a paired nonparametric bootstrap over problem IDs with $10{,}000$ resamples. Every selector is recomputed on the same resampled problem multiset, preserving the fixed-rollout pairing; percentile 2.5% and 97.5% quantiles form the reported 95% intervals. For split-sample packing checks, the proxy is estimated from the first 16 of 32 GSM8K candidates or first 8 of 16 MATH candidates, and all outcomes are evaluated on the disjoint remainder.

MATH answers are scored by final-answer equivalence, matching the fixed-rollout evaluation protocol used for GSM8K. MBPP public tests are treated as a practical selector channel, with correctness labels used only for evaluation. For the MBPP corruption diagnostic, we reduce the public-test signal before selection and then evaluate the selected program against the correctness criterion. This mirrors the corrupted-oracle intervention: the candidate set is fixed, and only the verifier channel used for selection is varied.

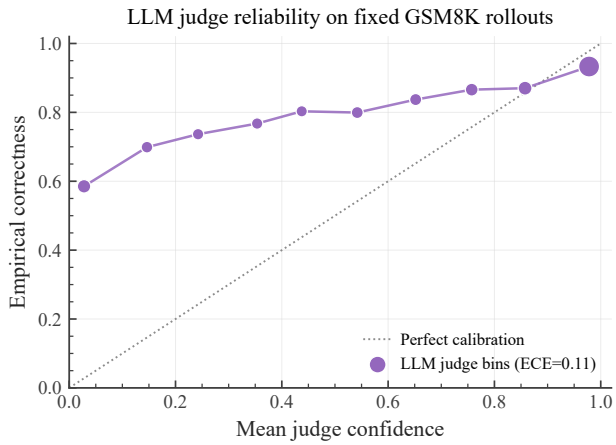

Figure 6: **LLM judge reliability diagnostic.** We bin candidate solutions by the Qwen2.5 judge's correctness probability and compare mean confidence to candidate correctness. The curve provides a reliability view of the judge signal, supporting the capacity-equivalent interpretation in the main text.

Table 3: Cross-family replication with Llama-3.1-8B-Instruct. Entries are captured fraction of the oracle gain at $N = 16$ (selected accuracy in parentheses, percent). Among the corrupted-oracle channels with prescribed capacity, η=0.2 exceeds η=0.4 in every column. The trained PRM (scored on the identical MATH rollouts, mean step reward) sits above every practical selector.

Selector (C_V bits)	MATH (250)	ARC-C (299)	OpenBookQA (500)
Pass@1	– (62.8)	– (81.6)	– (83.8)
Oracle (1.000)	100% (92.4)	100% (95.3)	100% (98.6)
PRM (Qwen2.5-Math-PRM-7B)	67.6% (82.8)	–	–
Noisy oracle η=0.2 (0.278)	50.0% (77.6)	39.0% (87.0)	52.7% (91.6)
Majority (n/a)	54.1% (78.8)	29.3% (85.6)	28.4% (88.0)
LLM judge (Qwen2.5-7B)	31.1% (72.0)	–	–
Noisy oracle η=0.4 (0.029)	20.3% (68.8)	0.0% (81.6)	20.3% (86.8)

Table 4: Accuracy bucketed by the estimated per-problem packing size $\log_2 \hat{M}_{\text{eff}}$, at $N = 32$ on GSM8K (Qwen2.5-7B) and $N = 16$ on MATH (Llama-3.1-8B). In both families, capacity-limited selection loses the most accuracy where $\log \hat{M}_{\text{eff}}/C_V$ is large.

	GSM8K (Qwen2.5-7B)				MATH (Llama-3.1-8B)			
$\log_2 \hat{M}_{\text{eff}}$	Ex.	Oracle	η=0.2	η=0.4	Ex.	Oracle	η=0.2	η=0.4
≤ 0.5	394	98.9	98.2	97.9	104	98.1	97.2	96.1
0.5–1	53	98.1	88.7	81.0	48	93.8	83.2	73.8
1–2	39	97.4	77.0	62.0	35	91.4	74.7	59.5
> 2	14	78.6	53.8	35.7	63	82.5	41.2	23.2

Table 5: Split-sample check for $\hat{M}_{\text{eff}}$. We estimate the packing proxy from the first half of each rollout set and evaluate the remaining candidates. "Loss" is held-out oracle accuracy minus the conditional expected accuracy of the η=0.4 selector. The gradient therefore does not reuse the candidates that estimated the proxy.

	GSM8K (16 estimate / 16 held out)			MATH (8 / 8)		
$\log_2 \hat{M}_{\text{eff}}$	Ex.	Oracle	Loss	Ex.	Oracle	Loss
≤ 0.5	388	98.5	0.6	98	96.9	4.7
0.5–1	65	98.5	19.0	49	93.9	14.0
1–2	33	87.9	33.4	53	79.3	22.2
> 2	14	71.4	44.8	50	68.0	38.6

Table 6: Full-suite multiple-choice summary at $N = 16$. "Oracle" is best-of-16 on the same fixed rollouts; "Majority" is self-consistency over the answer strings. The oracle gain column measures latent correct candidates available to a sufficiently informative selector. ARC-Easy's 0.5-point gain is within sampling uncertainty and is not treated as positive evidence.

Dataset	Examples	Pass@1	Oracle	Majority	Oracle gain
ARC-Challenge	299	87.6	90.0	89.0	2.3
ARC-Easy	570	92.3	92.8	92.1	0.5
BoolQ	1000	81.3	84.1	81.3	2.8
CommonsenseQA	1000	82.3	85.2	81.9	2.9
HellaSwag	1000	68.8	73.9	68.8	5.1
MMLU	1000	72.9	81.1	73.1	8.2
OpenBookQA	500	83.4	87.2	84.4	3.8
PIQA	1000	86.5	90.4	86.9	3.9
QASC	926	80.8	84.1	80.0	3.3
RACE	1000	86.1	87.6	85.6	1.5
SciQ	1000	94.2	95.6	94.1	1.4
SocialIQA	1000	73.8	78.3	73.5	4.5
WinoGrande	1000	70.6	75.3	69.7	4.7
Average	–	81.6	85.0	81.6	3.5