Efficient Thinking III: Efficient Judging

When an LLM judge is worth its compute — and the two conditions it must clear

Louay Alsakka · July 17, 2026 · *draft v0.1*

Abstract

After a model samples N candidate answers, something must select one. Three selectors exist: a verifier, exact but available only in checkable domains; majority vote, free wherever answers can be counted; and an LLM judge, available everywhere and costing compute. Papers I–II of this series measured what search extracts and what evaluators bound; this paper prices the judge. We ran a 48-cell grid — six judge sizes (1.5B–72B) against four policy sizes (0.5B–7B), at N = 4 and N = 16 candidates over frozen GSM8K caches, 150 problems per cell — in two protocols (pick-best-from-list and pairwise tournament), with every claimed effect tested by exact paired McNemar.

Judges earned wins in exactly one corner. Eight cells survive significance, all of them a strong judge (14B–72B) selecting for a weak policy (0.5B–1.5B), and six of the eight at N = 4. Every apparent win for a competent policy (3B–7B) is statistically null (p = 0.22–1.0) — on GSM8K. A pre-registered cross-benchmark test then showed the criterion predicting where its own answer changes: on MATH, where the 3B policy's consensus falls from 86% to 52%, the same strong judges win significantly (+4.0 and +4.6 at N = 4, corrected p = 0.024 and 0.025), while every such win remains FLOP-dominated by a bigger policy with the free vote. The pattern follows a two-factor selection–coverage criterion inherited from Paper II's proposition: a judge beats the vote only if its selection accuracy among covered problems exceeds the policy's mode-correctness — a condition that held with zero counterexamples across all 48 cells — and only where coverage admits a better answer to select, which gates the 16 cells where a competent judge still lost. Judge value = selection × coverage, the same decomposition, measured a third way.

Two further results price the judge honestly. First, allocation: every significant judge win is FLOP-dominated by a judge-free configuration — a bigger policy with free majority vote reaches higher accuracy at a fraction of the cost — so on this benchmark, under the budgets measured, the optimal share of compute spent on judging is approximately zero wherever a vote is available, and our registered predictions of an interior optimum are scored as misses. Second, the N tax: six of the eight wins are at N = 4, and at N = 16 only the two strongest-judge cells on the 1.5B policy survive in pick-best (one marginally, p = 0.043); in pairwise, nothing

survives at N = 16. Pick-best drowns by position — the 72B judge chose the first or last candidate in 52 of 60 trials against a uniform expectation of 15, blind to the middle of its own list — and pairwise tournaments compound elimination errors round by round. Below the competence floor a judge is worse than free: the 1.5B and 3B judges win nowhere and are significantly destructive in 8 of 16 pairwise cells. One sentence: pay for a judge only when the world gives you neither a vote nor a verifier — and then only a judge that clears the consensus it replaces, over a field small enough to actually read. All experiments ran on two Apple-Silicon machines over Paper II's frozen caches; claims are scoped to this benchmark and these budgets.

Results at a glance

finding	measurement	where
judges win in one corner only	8/48 cells significant, all weak-policy × strong-judge; every competent-policy "win" null	§3
the selection–coverage criterion	selection condition necessary in 48/48 GSM8K + 14/14 MATH cells (zero counterexamples); coverage gates the predicted-only cells	§3
the criterion predicts where its own answer changes	consensus 86% → 52% flips null cells to significant wins (corrected p = 0.024/0.025); s ≈ 0 stands regardless	§3
judge compute is never the right marginal spend here	each significant win FLOP-dominated by a judge-free config at higher accuracy	§4
the N tax	six of eight wins at N = 4; at N = 16 only 32B/72B judges on the 1.5B policy survive (pick-best), nothing in pairwise	§5
the mechanism, photographed	52/60 choices at list edges (uniform ≈ 15); correct answers missed uniformly in the middle	§5
below the floor, worse than free	weak judges: zero wins, 8 significantly negative cells, worst -20.0	§5

1. Introduction

This paper shows that an LLM judge is worth compute only when two measurable conditions hold — selection quality above the free consensus, and coverage that leaves something better to select — and that otherwise the judge's FLOPs are better spent on the policy itself. Everything that follows is evidence for that sentence.

Paper II ended at a gap: over identical samples, a perfect verifier kept climbing while majority vote saturated, and an LLM judge captured part of the difference. That result priced nothing —

it showed a judge *can* select better than a vote without saying when the judge's own compute is worth spending. This paper asks the priced question. At a fixed budget, a system designer chooses how to split compute between the policy that generates candidates and the judge that selects among them; call the judge's share s. The proposal for this paper registered six predictions about that split before any cell ran, including an interior optimum (P1) and a regime where a smaller policy with a larger judge beats the reverse (P5). The grid falsified both, and the falsifications are the paper's most useful content: on a benchmark where majority vote works, s ≈ 0 is optimal everywhere we measured, and the conditions under which a judge earns any compute at all turn out to be exactly two, both measurable in advance.

Contributions, in the order the paper develops them: the 48-cell judge-policy grid with per-cell exact significance, locating all judge value in the weak-policy × strong-judge corner (§3); the selection–coverage criterion — selection above consensus as a necessary condition with zero counterexamples, coverage as the gate on sufficiency — connecting judging to Paper II's mode-versus-coverage proposition cell by cell (§3); the FLOP-dominance result scoring the registered allocation predictions as misses (§4); the N tax with its positional mechanism photographed, and the protocol taxonomy that follows (§5); and the scored-prediction ledger, which this time includes two analysis errors caught during the paper's own review (§7).

A note on statistical power, stated once. Every cell is n = 150 problems on GSM8K over Paper II's frozen 32-sample caches (sample once, select many); every claimed win or loss carries an exact paired McNemar p-value, and cells that do not reach $p < 0.05$ are called null regardless of their point estimate. This discipline deletes ten of the eighteen apparent judge wins. Single-benchmark scope is the paper's main limitation and §8 registers the direction of the expected difference on harder benchmarks.

2. The grid

Six judges (Qwen2.5-Instruct 1.5B, 3B, 7B, 14B, 32B, 72B, 4-bit) select among N ∈ {4, 16} candidates drawn from four policies (0.5B, 1.5B, 3B, 7B) on 150 GSM8K problems, against three references computed from the same samples: majority vote (free), oracle best-of-N (coverage), and the policy's greedy answer. Two protocols: **pick-best**, one call with all N candidates in context; **pairwise**, a single-elimination tournament of two-candidate calls. Token counts are logged per cell (mean generation tokens per candidate; mean judge input tokens), so every configuration carries its compute cost in params × tokens units. The full grid at both N, pick-best mode (**bold** = significant win over majority):

judge \ policy	0.5B	1.5B	3B	7B
1.5B judge 24.7 / 25.3	53.3 / 60.0	68.7 / 75.3	88.0 / 90.7	
3B judge 24.0 / 25.3	56.7 / 60.0	72.0 / 75.3	87.3 / 90.7	
7B judge 31.3 / 25.3	62.7 / 60.0	71.3 / 75.3	89.3 / 90.7	

judge \ policy	0.5B	1.5B	3B	7B
14B judge **33.3** / 25.3	**68.0** / 60.0	76.0 / 75.3	88.7 / 90.7	
32B judge **36.7** / 25.3	**66.7** / 60.0	74.0 / 75.3	92.0 / 90.7	
72B judge **36.7** / 25.3	**71.3** / 60.0	75.3 / 75.3	93.3 / 90.7	
(oracle @N=4) 46.7	*77.3*	*86.7*	*96.0*	

judge \ policy	0.5B	1.5B	3B	7B
1.5B judge 25.3 / 36.7	56.0 / 64.0	58.7 / 86.0	86.7 / 92.7	
3B judge 24.0 / 36.7	56.7 / 64.0	58.0 / 86.0	89.3 / 92.7	
7B judge 28.7 / 36.7	66.0 / 64.0	67.3 / 86.0	85.3 / 92.7	
14B judge 35.3 / 36.7	69.3 / 64.0	72.7 / 86.0	90.0 / 92.7	
32B judge 42.7 / 36.7	**74.0** / 64.0	76.0 / 86.0	94.0 / 92.7	
72B judge 44.0 / 36.7	**71.3** / 64.0	70.7 / 86.0	86.7 / 92.7	
(oracle @N=16) 69.3	*92.0*	*94.7*	*96.7*	

3. Where judges win, and the criterion that says so — Anchor 1

Applying the no-win-without-significance rule to the grid leaves exactly eight cells:

judge	policy	N	Δ (pick − majority)	exact p
14B	0.5B	4	+8.0	0.0042
14B	1.5B	4	+8.0	0.0118
32B	0.5B	4	+11.4	0.0005
32B	1.5B	4	+6.7	0.0129
32B	1.5B	16	+10.0	0.0041
72B	0.5B	4	+11.4	0.0009
72B	1.5B	4	+11.3	0.0002
72B	1.5B	16	+7.3	0.0433

All eight are a strong judge selecting for a weak policy, and six of eight are at N = 4. The ten remaining positive cells — including every cell where a judge appears to beat a competent 3B or 7B policy — are statistically null (p = 0.07–1.0), and this paper calls none of them wins.

The corner is not an accident of scale but of arithmetic. Define q_judge as the judge's selection accuracy restricted to problems where a correct answer is present among the candidates, and mode-correctness as the policy's majority-vote accuracy — the strength of the free consensus the judge must beat. Across all 48 cells, **the condition q_judge > mode-correctness is necessary without exception**: 0 cells beat majority while failing it. It is not sufficient: 16 cells clear it and lose anyway, and each of those is gated by coverage — the judge selects well among the answers present, but too few problems have a correct answer present for good selection to overcome a strong vote. Judge value is therefore the product of two measurable factors, selection margin and coverage, which is Paper II's proposition — majority converges to the mode, oracles to coverage — operating inside a third experiment. The 2×2 over the grid: predicted-and-won 18, predicted-and-lost (coverage-gated) 16, unpredicted-win 0, neither 14.

The criterion's practical form: both factors are computable *before* deploying a judge — mode-correctness from a small sample of votes, q_judge from a small labeled calibration set — so whether a judge will pay is a measurement, not a bet.

The criterion predicting where its own answer changes (MATH)

The sharpest test of a criterion is whether it forecasts the conditions under which its own verdict flips, so we registered one before running it: on MATH, where the 3B policy's consensus collapses from 86% to 52%, the strong judges that were null on GSM8K should become significant wins at N = 4, while the allocation conclusion should survive. The 14-cell MATH grid (judges 7B–72B × policies 3B/7B, n = 300, scored under an analysis plan frozen before the decisive cells completed):

judge	policy	N	Δ (pick − majority)	exact p
7B	3B	4	-1.4	0.5235
7B	3B	16	-14.0	0.0000
14B	3B	4	-0.4	1.0000
14B	3B	16	-10.0	0.0000
14B	7B	4	+2.0	0.2379
14B	7B	16	-3.7	0.0347
32B	3B	4	+4.0	0.0118
32B	3B	16	-3.4	0.1214
32B	7B	4	+2.6	0.0963

judge	policy	N	Δ (pick − majority)	exact p
32B	7B	16	-1.0	0.6776
72B	3B	4	+4.6	0.0125
72B	3B	16	-3.4	0.1641
72B	7B	4	+2.6	0.0963
72B	7B	16	-2.4	0.2100

The registered cells landed: 32B×3B at +4.0 and 72B×3B at +4.6, both significant under the plan's Bonferroni correction (adjusted p = 0.024 and 0.025; both also survive the stricter three-way correction), while the pre-designated 14B boundary cell landed null, as the plan allowed. The criterion's necessity stayed exceptionless on the new benchmark (0 violations in 14 cells), the N = 16 column stayed entirely non-positive (0/7 positive), and every new win remains FLOP-dominated by a bigger policy with the free vote (the best judge win reaches 45.3 where plain 7B majority reaches 69.7 at a fraction of the cost) — the judge-viable region widens with weaker consensus, exactly as the criterion prescribes, and the judge-optimal allocation still does not appear. One further sharpening came free: the 7B and 14B judges are not merely taxed at N = 16 on MATH but significantly destructive (−14.0 and −10.0, $p < 0.0001$), despite the 7B judge being serviceable on GSM8K's weak policies — the competence floor is task-relative, and it moves up with the difficulty of the judging task itself.

4. The price: allocation, scored — Anchor 2

The proposal's central registered prediction (P1) was an interior optimum: some split s between policy and judge compute beating both endpoints. The grid says otherwise. Pricing each significant judge win against judge-free configurations from the same caches (cost in params × tokens per problem):

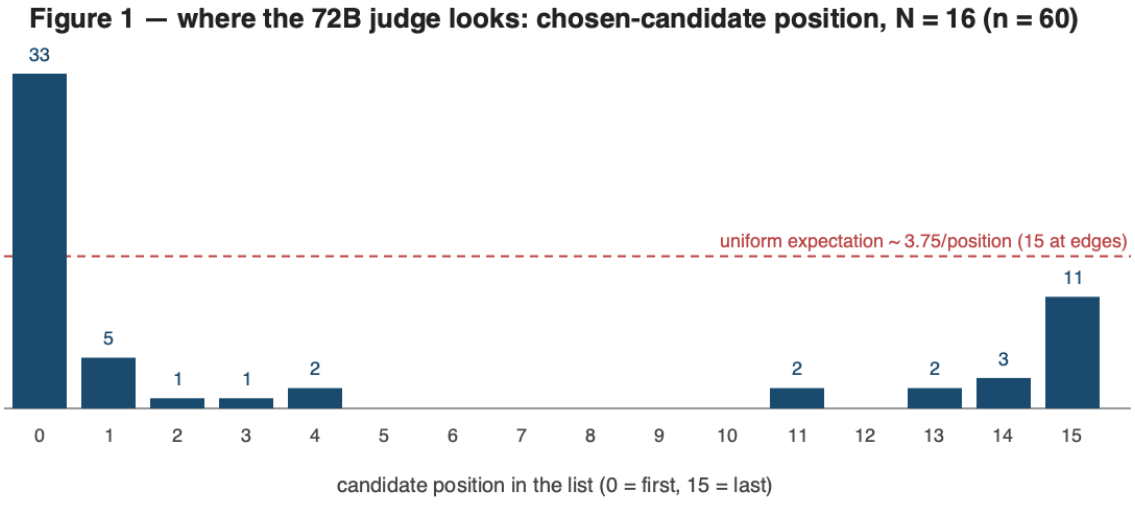
judge win (cell)	acc	cost	judge-free config	acc	cost	win costs
14B×0.5B N=4	33.3	29k	1.5B+majority@4	60.0	2k	17.9×
14B×1.5B N=4	68.0	19k	3B+majority@4	75.3	4k	4.9×
32B×0.5B N=4	36.7	64k	1.5B+majority@4	60.0	2k	40.1×
32B×1.5B N=4	66.7	41k	3B+majority@4	75.3	4k	10.6×
32B×1.5B N=16	74.0	151k	3B+majority@4	75.3	4k	39.0×
72B×0.5B N=4	36.7	144k	1.5B+majority@4	60.0	2k	89.4×
72B×1.5B N=4	71.3	90k	3B+majority@4	75.3	4k	23.2×

judge win (cell)	acc	cost	judge-free config	acc	cost	win costs
72B×1.5B N=16	71.3	332k	3B+majority@4	75.3	4k	85.7×

Every significant judge win is dominated — a judge-free configuration reaches strictly higher accuracy at a fraction of the cost, because policy FLOPs buy coverage and consensus simultaneously while judge FLOPs buy only selection over a fixed, weak candidate pool. P1 (interior optimum) and P5 (smaller policy + larger judge beats the reverse) are scored as misses; P2 (judge share grows with budget) inherits the miss, since $s \approx 0$ at every budget we can construct from these cells. The exception that defines the judge's real niche is a deployment lock: when the policy is fixed — by memory, latency, or product constraints — and its consensus is weak, a strong judge is the best selector available, and the eight-win corner is exactly that regime. Paper II found the same structure for search itself: the lever pays on the constrained frontier, not the free one.

5. The N tax: two protocols, two failure modes, one mechanism

Growing the candidate field is a tax in both protocols, paid unevenly. Of the eight significant wins, six are at N = 4; at N = 16 only the two strongest judges (32B, 72B) keep a significant win, and only on the 1.5B policy — the 72B cell marginally ($p = 0.043$). In pick-best, accuracy falls outright as the list grows for weaker judges: the 3B judge drops 14 points from N = 4 to N = 16 on the 3B policy, landing 28 below the free vote it was hired to beat. The mechanism is positional. Instrumenting the 72B judge's choices at N = 16:



The judge chose an edge of the list 52 times in 60 (uniform expectation ≈ 15), overwhelmingly the first position, while the correct answers it missed were distributed uniformly through the middle — lost-in-the-middle [Liu et al. 2023], photographed in the act of deleting a judge's value. Sixteen candidates in context is effectively a choice among two.

Pairwise removes the list and pays a different tax. Re-running the two cells where the criterion predicted wins that pick-best failed to deliver (registered as R-a), with the N = 8 midpoint added:

judge	N	Δ pairwise − majority	exact p
7B	4	+8.7	0.0106
7B	8	-4.7	0.2962
7B	16	-1.4	0.8555
14B	4	+9.4	0.0066
14B	8	+6.0	0.0931
14B	16	+4.6	0.3105

Both judges win significantly at N = 4; nothing survives at N = 16 — the registered prediction that pairwise would flip the N = 16 cells to wins is scored a miss. Pairwise recovers most of the drowned deficit (+6 to +7 points over pick-best at N = 16) but compounds elimination error: each round is a fresh chance to discard the correct answer permanently, and the rounds grow with N. The residual margins order by the judge's selection gap over consensus, exactly as the criterion prescribes: the 14B judge (larger margin) ends positive-null, the 7B judge (marginal) at parity.

Below the competence floor, no protocol helps and the judge is actively harmful: across 16 pairwise cells, the 1.5B and 3B judges win nowhere, are significantly *worse* than free majority in 8 cells, and bottom out at -20.0 points (1.5B judge, 3B policy, N = 16). A weak judge does not merely waste its compute; it converts a working vote into a worse decision.

The protocol decision rule that falls out: keep candidate fields small (the value concentrates at N = 4; only the largest judges on the weakest consensus retain N = 16 wins, and only in pick-best); use pick-best for its single-call cheapness when the judge clears the criterion with margin; use pairwise only to rescue a strong judge from a list it would drown in; and never deploy a judge below the competence floor, where the correct protocol is no judge at all.

6. Synthesis, and the handoff

Assemble the pieces and the judge's contract is short. A judge is worth compute when — and only when — (1) no verifier exists, (2) the free vote is weak (low mode-correctness), (3) the judge's selection accuracy clears that vote (the criterion's necessary condition), (4) coverage admits something better to select, and (5) the candidate field is small enough to read. On GSM8K, a checkable task with strong consensus, conditions (1) and (2) already fail for every competent policy, and the marginal FLOP always belonged to the policy. That is this paper's negative result, and its value is the domain it points at: creative and aesthetic work, where

there is no verifier, no meaningful vote over poems, and the judge is not an option but the only evaluator there is. Paper IV begins exactly there, carrying this paper's machinery — the pairwise protocol, small fields, the calibration-first measurement of q — into the regime where the judge is forced.

Related work

The LLM-as-a-judge literature [Zheng et al. 2023; Arena-Hard] measures how well judges agree with human or ground-truth rankings; this paper asks a different question — whether the judge's selection is worth its compute against the free baselines a deployed system already has, and finds the answer is a measurable condition, not a benchmark score. The best-of-N and verifier line [Cobbe et al. 2021; Lightman et al. 2023; Brown et al. 2024] established that selection quality gates repeated sampling; our contribution is the pricing of the intermediate case, a learned judge against a free vote, and the exceptionless necessity of the selection condition across a judge–policy grid. Position bias in list-based judging is documented qualitatively [Zheng et al. 2023; Wang et al. 2023]; Figure 1 prices it — the collapse converts a significant N = 4 win into an N = 16 null even for the largest judge, at both 4-bit and 8-bit precision and out-of-family (Llama-8B, 48/60 edges). The allocation question — policy versus selector compute at fixed budget — is to our knowledge unpriced in this form; the answer here ($s \approx 0$ on a consensus-strong benchmark, under the budgets measured) is scoped in §8 and registered for a consensus-weak benchmark where the criterion predicts it changes.

7. Registered predictions, scored

prediction (registered before the run)	outcome
P1 — an interior allocation optimum exists for mid-capability policies	**Miss.** $s \approx 0$ everywhere measured; every significant judge win is FLOP-dominated by a judge-free configuration (§4)
P2 — optimal judge share grows with budget	**Miss, inherited.** With $s \approx 0$ at all constructible budgets, no growth exists to observe (§4)
P3 — judge size beats judge search at equal FLOPs	**Unscored.** Judge-side self-consistency cells were not run; deferred to the ET-IV harness
P4 — q(j) rises smoothly with a competence knee	**Hit at N = 4, protocol-confounded at N = 16.** Mean q_judge at N = 4: 1.5B 73%, 3B 75%, 7B 81%, 14B 85%, 32B 86%, 72B 89% — smooth, with the collapse below 7B; at N = 16 the 72B point inverts, an artifact of position collapse (§5), not of judge quality

prediction (registered before the run)	outcome
P5 — smaller policy + larger judge beats the reverse	**Miss.** Policy FLOPs dominate wherever the comparison can be built (§4)
P6 — pick-best beats pairwise at fixed judge FLOPs	**Split by regime.** Pick-best wins for mid judges (pairwise compounds their errors); pairwise wins for strong judges on weak policies at large N (it removes the drowning tax); neither survives N = 16 (§5)
Absolute competence knee at ~7B: judges beat majority from the same threshold as ET-II's crossover (in-conversation, pre-grid)	**Miss.** No judge size wins across policies; even 72B loses to a competent policy's consensus. Value is relative, not absolute (§3)
Relative competence: the 3B judge helps weak policies, loses to strong ones (in-conversation, pre-grid)	**Partial miss.** Direction right, floor wrong: the 3B judge helps nobody and is significantly harmful below the floor (§5)
N-degradation shrinks with judge size (in-conversation, pre-grid)	**Partial hit.** Monotone easing from 1.5B to 32B; the 72B reversal at N = 16 resolved by the position diagnostic as protocol, not capability (§5)
R-a — pairwise flips the two criterion-mismatch cells to wins at N = 16	**Miss as registered; mechanism confirmed.** Pairwise recovers the drowned deficit into parity, and both cells win significantly at N = 4; the N tax, not the list alone, is what binds (§5)
Negative control — weak judges win nowhere, even in pairwise	**Hit,** with a bonus severity: 8 significantly negative cells (§5)
L1 — the criterion transfers across model family (Llama-8B judge)	**Hit.** Necessity exceptionless in all six cells; the failure structure (coverage-gated at N = 4, drowning at N = 16) matches the mid-size Qwen judges quantity for quantity (§8)
L2 — position collapse replicates out-of-family	**Hit.** 48/60 edge choices against a uniform expectation of 15 (§8)
M1 — strong judges, null on GSM8K, become significant wins on MATH's weaker consensus	**Hit.** +4.0 and +4.6 at N = 4, both clearing the frozen plan's correction (and the stricter ×3); the pre-designated 14B boundary cell null, as allowed (§3)
M2 — the criterion's necessity holds on MATH	**Hit.** Zero violations in 14 cells (§3)
M3 — the allocation conclusion survives: new wins remain FLOP-dominated	**Hit.** Best judge win 45.3 vs plain 7B majority 69.7 at a fraction of the cost; the region widens, $s \approx 0$ stands (§3)

prediction (registered before the run)	outcome
M4 — the N tax persists cross-benchmark	**Hit,** with a sharpening: N = 16 entirely non-positive, and the mid judges significantly destructive (−10 to −14, $p < 0.0001$) — the competence floor is task-relative (§3, §5)

Two analysis errors were caught during this paper's own review and belong in its record: an aggregate form of the criterion was briefly computed with a shared denominator, making it a tautology (24/24 agreement by identity), and a units mismatch then inverted it; both were caught before any conclusion rested on them, and the per-problem form reported in §3 is the corrected computation, independently reimplemented against the committed per-problem files. The program's rule — no delta believed until it survives a low-variance re-measurement — applied to its own instruments twice this cycle.

8. Limitations and future work

One benchmark: GSM8K's consensus is unusually strong (high mode-correctness), which is precisely the condition that starves judges; the cross-benchmark question is now answered rather than registered: the MATH judge grid (§3) confirmed the criterion's forecast in full — the judge-viable region widens with weaker consensus, necessity stays exceptionless, and the allocation conclusion survives — so the paper's scope claim upgrades from one benchmark to two, with the criterion itself as the bridge. The grid is one model family, and the judge-side family check has now been run: a Llama-3.1-8B judge over the same caches obeys the selection–coverage criterion without exception (six cells, zero violations) and fails in the same structure as the mid-size Qwen judges — it clears the selection bar over weak consensus by a wide margin ($q = 0.578$ against a 0.250 consensus) yet lands null because coverage gates it, and it drowns significantly at N = 16 (−7.0, $p = 0.013$; −15.0, $p < 0.001$). The criterion is about quantities, not family. The position-collapse result has likewise been checked twice over: the edge-choice rate persists at higher precision (52/60 at 4-bit, 55/60 at 8-bit) and out-of-family (Llama-8B, 48/60), so Figure 1 reads as a property of long-list judging itself. The policy-side family sweep remains open. n = 150 per cell bounds the detectable effect at roughly ±7 points; the corner wins clear it, the competent-policy nulls may hide real ±2-point effects we cannot see. Judge prompts were fixed and unoptimized; prompt engineering the judge is a confound we priced at zero and a reader should not. P3 remains unscored and moves to the ET-IV harness, where judge-side compute is the entire subject.

Reproducibility

All grids, per-problem outcome files, pairwise and diagnostic runs, and the analysis scripts (including the bridge computation and this draft's build script) are in the repository under

`judging/`, generated over Paper II's frozen caches (`reasoning/`). Every number in this paper is computed from those files by script at build time; none is transcribed.

References

- Alsakka, L. (2026). *Efficient Thinking I: Measuring What Capability Costs*. This series.
- Alsakka, L. (2026). *Efficient Thinking II: Where Search Pays and Where It Can't*. This series.
- Brown, B. et al. (2024). *Large Language Monkeys: Scaling Inference Compute with Repeated Sampling*. arXiv:2407.21787.
- Cobbe, K. et al. (2021). *Training Verifiers to Solve Math Word Problems*. arXiv:2110.14168.
- Li, T. et al. (2024). *From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard*. arXiv:2406.11939.
- Lightman, H. et al. (2023). *Let's Verify Step by Step*. arXiv:2305.20050.
- Liu, N. F. et al. (2023). *Lost in the Middle: How Language Models Use Long Contexts*. TACL.
- Wang, P. et al. (2023). *Large Language Models are not Fair Evaluators*. arXiv:2305.17926.
- Wang, X. et al. (2022). *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. arXiv:2203.11171.
- Zheng, L. et al. (2023). *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. NeurIPS.