

Review Rarely Beats the Stronger Model: Separating Verification from Capability Transfer in LLM Pipelines

Yirui Zhang Yash Raj Shrestha

University of Lausanne

{yirui.zhang, yashraj.shrestha}@unil.ch

Abstract

Qwen3-8B reviewing Llama-3.1-8B in math adds 33.2 points, yet Qwen3-8B alone is 6.4 points more accurate than the result. Reviewing another model’s answer is usually judged by its gain over the generator. We show that this confuses verification, which adds accuracy beyond either model alone, with capability transfer, where a stronger reviewer supplies answers it could have given on its own. Measured against the stronger model alone, across 48 pairs of 12 open-weight models (3B to 72B) in math, knowledge, code and reasoning, review beats it in only 9 of 96 review directions, and does worse the larger the capability gap. The reason is that the reviewer mostly follows the generator, discarding its own correct answers. Review costs 2.6 to 2.8 times as much as the stronger model alone, while a selector using the same two calls is 2.7 to 4.2 points more accurate than review and cheaper. Studies of verification should report the stronger model alone, not only the gain over the generator. Code and per-question results: <https://github.com/Kamilaabb/verification-or-capability-transfer>.

1 Introduction

As LLM outputs outpace human inspection (Lin and Zhang, 2026), verification is increasingly delegated to models themselves (Zheng et al., 2023). Language model inference is costly in compute, money and energy (Luccioni et al., 2024), so a second model call is justified only if it buys accuracy that a single call cannot. A common way to spend it is review, as when coding agents check each other’s changes (Xiang et al., 2026) or a multi-agent reasoning system adds a reviewer to its solver (Yang et al., 2026). A generator (G) answers a question, and a reviewer (V) reads the question and G’s full response and returns the final answer, either keeping G’s answer or correcting it (Welleck et al., 2023; Xiang et al., 2026). Review, and LLM ver-

ification more broadly, is usually judged by how much the final answer improves on G answering alone (Welleck et al., 2023; Lifshitz et al., 2025; Zhou et al., 2026; Lu et al., 2026), and by that measure it often looks very effective: in our math experiments, Qwen3-8B (Yang et al., 2025a) reviewing Llama-3.1-8B (Grattafiori et al., 2024) raises Llama-3.1-8B’s accuracy by 33.2 points.

But review takes two model calls, and one of them already goes to the stronger model. Here the extra call buys less than nothing: Qwen3-8B answering alone is 6.4 points more accurate than the review’s final answer. A gain over G can come from two very different sources: verification, which adds accuracy beyond either model alone, and capability transfer, where a stronger reviewer supplies answers it could have given on its own. The standard comparison cannot tell them apart and credits both to review. This hides a waste: a pipeline may pay for review believing it verifies, while a single call to the stronger model would cost less and be at least as accurate, and research may count capability transfer as progress in verification. In this paper, we separate the two by measuring review in two ways (Figure 1a): the gain over the generator (Δ_G), the final answer’s accuracy minus G’s accuracy alone, and the gain over the stronger model (Δ_S), the final answer’s accuracy minus that of the more accurate of the two models answering alone.

A selector can only help where the second model is right and the stronger one wrong. We call the share of such questions the **headroom**: it is what an ideal choice between the two independent answers would add over the stronger model. Review should be able to collect it, since the reviewer sees both the question and another model’s attempt.

We measure this on 48 pairs of 12 open-weight models from 3B to 72B parameters, with 2,500 math, 2,500 knowledge, 832 code and 901 reasoning questions, all graded by program. For every pair we run review in both directions, stronger re-

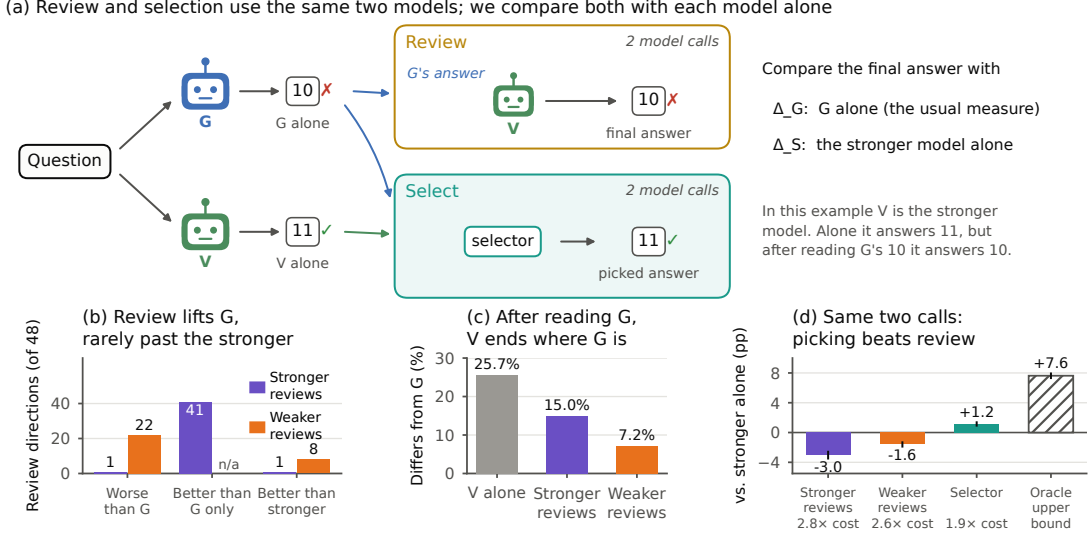


Figure 1: Overview. (a) In review, the reviewer (V) reads the generator’s (G) answer and returns the final answer; in selection, a selector picks one of the two models’ independent answers. Both use two model calls. We compare the final answer with G alone (Δ_G) and with the stronger model alone (Δ_S). (b) Outcomes of the 48 directions of each kind. When the stronger model reviews, review beats G in 42 but the stronger model in only one; when the weaker one reviews (G is stronger), it beats both in 8 and is worse than G in 22. (c) After review, the final answer differs from G’s in correctness far less often than V’s own answer did. (d) With the same two model calls, choosing between the two independent answers is more accurate than review; the oracle, which picks a correct answer whenever either model has one, shows how much a pair could offer.

viewing weaker and weaker reviewing stronger, 96 review directions in total. We choose pairs so that both the gap, the difference in accuracy between the two models answering alone, and the headroom vary widely. We ask what review does (RQ1) and why (RQ2), whether a second model is worth its cost (RQ3), and whether to select or review (RQ4).

Our experiments show that review rarely beats the stronger model answering alone (Figure 1b): only 9 of 96 directions do, seven of them a weaker model reviewing a close stronger one, and the larger the gap between the two models, the worse review does. The reason is that the reviewer follows G: on questions that the reviewer answers correctly alone and G gets wrong, the final answer is right only 40% of the time when the stronger model reviews and 17% when the weaker one does (Figure 2). After review, the final answer differs from G’s in correctness far less often than the two models alone do (Figure 1c), so most of the 7.5 points of headroom a pair leaves on average is lost. With the same two calls, a simple selector that picks one of the two independent answers is 4.2 and 2.7 points more accurate than the two review directions and costs less in every pair (Figure 1d).

In summary, our key contributions are: 1) We reveal that judging review by its gain over the gen-

erator confuses verification with capability transfer; 2) By splitting questions by which model is right alone, we show that review falls short because the reviewer follows the generator’s answer; 3) We propose an evaluation framework with a second reference, a taxonomy of review outcomes and an equal-call baseline that any review method should beat, and find that even a simple selector beats review at lower cost.

2 Related Work

Review and verification. We study zero-shot review by general-purpose models: a second model reads another model’s full answer and returns the final one. Trained correctors and verifiers (Welleck et al., 2023) are out of scope. A second model is a natural choice, since without external feedback models struggle to correct their own reasoning (Kamoi et al., 2024) and can even lose accuracy (Huang et al., 2024), and verifying a solution is often thought to be easier than generating one (Zhou et al., 2026). Table 1 compares the closest studies. Most let the second model approve or reject candidates sampled from the generator and measure the gain over the generator (Lifshitz et al., 2025; Lu et al., 2026; Zhou et al., 2026). In this

Work	Second model’s role	Final answer	Reference (as defined in the paper)	Stronger model alone	Equal-call choice	Split by who is right alone
Huang et al. (2024)	Same model revises its own answer	Same model	Accuracy after vs. before self-correction	n/a	✓	n/a
Singhi et al. (2025)	Same model scores its own solutions	G’s sample	Success rate vs. self-consistency at equal compute	n/a	✓	n/a
Lifshitz et al. (2025)	Verifiers approve G’s samples; most approved is kept	G’s sample	Accuracy of BoN-MAV vs. pass@1 of G	when G is stronger	✓	×
Lu et al. (2026)	Verifier accepts or rejects; G resamples	G’s sample	$\text{Precision}(S, V) - \text{SolverAcc}(S)$	×	×	×
Zhou et al. (2026)	Verifier filters G’s samples	G’s sample	$\Delta p_V = p_{G,V} - p_G$	one case	×	×
Xiang et al. (2026)	Other model reviews and rewrites	Reviewer	Pass rate vs. each model alone	one pair	×	×
Ours	Other model reviews and rewrites	Reviewer	Δ_G and Δ_S (Section 3.1)	✓	✓	✓

Table 1: Closest prior work. Most studies let the second model choose among the generator’s own samples and measure the gain over the generator; in review, the second model writes the final answer. “Equal-call choice”: a baseline that picks among independent answers at the same cost. “Split by who is right alone”: outcomes split by whether each model answers correctly on its own.

Question

How many natural numbers less than 1000 have exactly three distinct positive integer divisors? (answer: 11)
 Solution: these are the squares of primes; $p^2 < 1000$ gives $p \leq 31$, and 11 primes are ≤ 31 .

Llama-3.1-8B answers (G, weaker) final answer 10 ✗
 [...] we need to find the prime numbers less than 31.
 The prime numbers less than 31 are 2, 3, 5, 7, 11, 13, 17, 19, 23, and 29. [...] There are 10 such numbers.

Qwen3-8B alone (V, stronger) final answer 11 ✓
 [...] 17, 19, 23, 29, 31 [...] $31^2 = 961$ [...] All of these are less than 1000.
 So, there are 11 such numbers.

Qwen3-8B reviews G’s answer (V) final answer 10 ✗
 Told: “The proposed solution may be correct or incorrect. [...] If the proposed solution is correct, retain it. If it contains any mathematical error, correct the error [...]”
 [...] we need to find all prime numbers less than 31.62, i.e., less than or equal to 31.
 The prime numbers less than or equal to 31 are: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29
 That’s a total of 10 primes.

Figure 2: A reviewer that solves a question alone follows the generator’s wrong answer after reading it. Llama-3.1-8B lists primes only up to 29 and answers 10. Qwen3-8B answers 11 (correct) on its own; reviewing, it writes the correct bound but copies the list that stops at 29, and answers 10.

form, several weak verifiers improve even stronger generators (Lifshitz et al., 2025): the generator is the stronger model, so this gain is also a gain over the stronger model, and it comes from choosing among answers, in line with our finding that choosing succeeds where review fails. Xiang et al. (2026) compare review with each model alone for

a single pair on coding: Claude reviewing Codex raises it from 71.6% to 89.7%, below Claude’s own 91.4%. None of these studies tests review against both the stronger model alone and an equal-call choice, or traces where its gains come from. We do both across 96 review directions in four domains, and show that review’s gains are mostly capability transfer.

Anchoring and conformity. Models are swayed by what they are shown: LLM judges by the order of candidates and by their own generations (Wang et al., 2024a; Zheng et al., 2023; Panickssery et al., 2024), and assistants by a user’s stated views (Sharma et al., 2024). Among peers, models follow a majority of scripted answers (Zhu et al., 2025; Weng et al., 2025), and agreeing peers mislead a model that answered correctly far more easily than they correct one that answered wrongly (Qu et al., 2026). In debates between real agents, agents defer to a peer more than they keep their own answer (Choi et al., 2026), and among models from 3B to 32B, smaller models had more influence on the others’ answer changes (Kasprova et al., 2026). These peers are scripted answers, copies of the same model or binary stances; in review, the peer is another model’s full solution, across real capability gaps in both directions. Review is also the minimal case: one model reads one other solution, with no user, no majority, and a note that the solution may be incorrect, and anchoring persists even here.

Prior work shows that models change answers under pressure; we show what this costs a pipeline: most of the headroom a pair offers is lost, which is why review loses to selection. Self-review shows that the cause is the other model’s answer, not the second pass: shown its own answer, a model keeps over 90% of its right answers (Appendix F).

Selection, routing and equal cost. Methods that use more than one model differ in when they act. Routers decide which model should answer before any answer exists (Ong et al., 2025; Ding et al., 2024; Shnitzer et al., 2024); cascades run a cheaper model first and call a stronger one when its output looks unreliable (Chen et al., 2024; Aggarwal et al., 2024; Li et al., 2026); ranking and fusion act after all candidates exist (Jiang et al., 2023). Review and selection both act after the answers exist: review shows one answer to the second model, while our selector, following selection-based ensembling (Jiang et al., 2023), picks one of the two independent answers. Comparisons at equal cost show that counting the cost of verification changes the conclusion: multi-agent debate is no better than self-consistency at the same number of responses (Huang et al., 2024), generative verification needs up to eight times the compute of self-consistency to match it (Singhi et al., 2025), and multi-agent verification overtakes both only with enough compute (Lifshitz et al., 2025). Commit-first judging has the judge write its own answer before using the candidate (Zhou, 2026; Gozel, 2026); in one call, with the candidate already in the input, we find that this answer is not independent of it (Appendix F). The equal-cost comparisons above choose among many samples of one generator. To our knowledge, no prior work uses choosing between two different models’ independent answers as the yardstick for review. Under it, review loses in both directions at the smallest budget, two calls: selection is more accurate on average, never significantly less accurate in any pair, and cheaper in every pair, with an external judge (Li et al., 2026) as a reference.

3 Framework and Experimental Setup

3.1 Review Protocol and Comparative Gains

Generation and Review. In each review direction, every question is answered three times. The two models of a pair each answer alone with the same prompt, and in a review one model (the generator, G) answers and the other (the reviewer, V) reads the question and G’s full response (Xiang

et al., 2026; Yang et al., 2026). V is told that the response may be right or wrong; it keeps the answer if it judges it correct and otherwise corrects it, and either way returns a complete solution whose answer is the final answer (GV; Appendix B). Each pair is run in both review directions: the stronger model reviewing the weaker, and the weaker reviewing the stronger. As sanity checks on a subset of questions, we also run a second review prompt, which drops the instruction to keep a correct answer and has V commit to its own answer first, and self-review, in which V reviews its own solo answer instead of G’s (Appendix F).

Notation. For question i of the n questions of a direction, $g_i, v_i, f_i \in \{0, 1\}$ indicate whether G alone, V alone and the final answer are correct, and s_i, w_i do the same for the stronger and the weaker model alone, ranked by accuracy on the development part (Section 3.2). A bar is a mean over questions, so $\bar{g}$ is G’s accuracy alone.

Gain over the generator and the stronger model.

$$\Delta_G = \bar{f} - \bar{g}, \quad \Delta_S = \bar{f} - \bar{s}. \quad (1)$$

Δ_G is the number most studies report; Δ_S asks whether review beats simply calling the stronger model. We call $\gamma = \bar{s} - \bar{w}$ the capability gap; it ranges from 0.5 to 39.6 points across our 48 pairs, and we report all differences in percentage points. Then $\Delta_G = \gamma + \Delta_S$ when the stronger model reviews and $\Delta_G = \Delta_S$ when the weaker one does: a large Δ_G for a stronger reviewer can come entirely from the gap.

Gap, disagreement, and headroom. Two more quantities describe the two independent answers. The disagreement $\delta = \frac{1}{n} \sum_i \mathbb{1}[s_i \neq w_i]$ is the share of questions that exactly one model gets right. The headroom

$$h = \frac{1}{n} \sum_i \mathbb{1}[w_i = 1, s_i = 0] \quad (2)$$

is what always picking a correct one of the two answers would add over the stronger model. Since $\delta = \gamma + 2h$, $h = (\delta - \gamma)/2$.

Decomposing review gains by question type. Grouping the questions by (g_i, v_i) splits the gain

Domain	Source	N	Grading
Math	MATH, levels 3–5	2,500	symbolic equiv.
Knowledge	MMLU-Pro	2,500	option letter
Code	BigCodeBench-Instruct	832	unit tests
Reasoning	ReClor, FOLIO v2, GPQA	901	option letter

Table 2: Tasks, from MATH (Hendrycks et al., 2021), MMLU-Pro (Wang et al., 2024b), BigCodeBench (Zhuo et al., 2025), and for reasoning 500 ReClor (Yu et al., 2020), 203 FOLIO v2 (Han et al., 2024) and 198 GPQA Diamond (Rein et al., 2024) questions.

over the generator into three parts:

$$\Delta_G = \underbrace{\frac{1}{n} \sum_{g_i=0, v_i=1} f_i}_{\text{V-only correct}} - \underbrace{\frac{1}{n} \sum_{g_i=1, v_i=0} (1 - f_i)}_{\text{G-only correct}} + \underbrace{\frac{1}{n} \sum_{g_i=v_i} (f_i - g_i)}_{\text{same outcome}} \quad (3)$$

The parts are the reviewer’s correct answers that the final answer takes up, G’s correct answers that review loses, and what review changes where the two models agree. The first part is capability transfer; only a positive Δ_S shows verification.

Taxonomy of Review Outcomes. From the paired 95% intervals I_G and I_S of Δ_G and Δ_S (Section 3.3), we place each direction at one of five outcome levels: worse than G ($I_G < 0$), no different from G ($0 \in I_G$), above G but below the stronger model ($I_G > 0, I_S < 0$), level with the stronger model ($I_G > 0, 0 \in I_S$), and beyond it ($I_S > 0$, which takes precedence).

3.2 Tasks and Models

All four domains are graded by program, never by an LLM judge (Table 2). In math and knowledge, 1,000 random questions form the development part used to rank models; other domains use all questions. An unparseable answer counts as wrong; if the reviewer’s answer cannot be parsed, or in code contains no program, the final answer is G’s.

Models and pairs. We use 12 open-weight instruction-tuned models from 3B to 72B parameters: Qwen3 (4B, 8B, 32B; Yang et al., 2025a), Qwen2.5 (3B, 7B, 72B; Yang et al., 2025b), Qwen2.5-Coder-7B (Hui et al., 2024), Seed-Coder-8B (ByteDance Seed et al., 2025), Llama-3.2-3B (Meta AI, 2024), Llama-3.1-8B (Grattafiori et al., 2024), Ministral-8B (Mistral AI, 2024) and Mistral-Small-24B (Mistral AI, 2025). Within each domain,

we sort the models into strong, middle and weak tiers by development accuracy and, for each of the six tier combinations, take one pair with high and one with low headroom, so that gap and headroom both vary widely (Appendix A). This gives 12 pairs, or 24 review directions, per domain.

All models decode greedily, with thinking off for Qwen3 and up to 4,096 output tokens, so each model has one fixed answer per question. Only 1.7–1.8% of review outputs and 1.8–3.3% of solo answers reach this limit (Appendix F). As a sanity check, we rerun Qwen3-8B with thinking on in two math directions (Appendix F). Prompts are in Appendix B.

3.3 Statistical Inference and Uncertainty

Our 95% intervals resample questions with replacement, keeping each question’s paired outcomes as one unit, so they measure the uncertainty from which questions were sampled; averages weight the four domains equally. To relate outcomes to the gap, we regress Δ_S and Δ_G on γ , alone and jointly with h , with one intercept per domain, resampling pairs as well, and read the slope as a linear association across the evaluated pairs (details in Appendix E).

4 Results

4.1 RQ1: What does review do?

By the paired 95% intervals, the 96 review directions fall into the five levels as follows: worse than G (23), no different from G (23), above G but below the stronger model (22), level with the stronger model (19), and beyond it (9). Which level a direction reaches is largely decided by who reviews whom and by the gap (Figure 3c; by gap range, Figure 6).

A stronger reviewer lifts G toward itself, not past it. Measured against the generator (Figure 3a), review looks very effective: the gain grows by 8.2 points per 10 points of gap [7.0, 8.9] and reaches 33.2 points when Qwen3-8B reviews Llama-3.1-8B in math. Measured against the stronger model (Figure 3b), the same directions fall below zero, on average 3.1 points [2.7, 3.5] below the reviewer answering alone, and 6.4 points for that math pair. The larger the gap, the more review lifts G and the further the result falls below the stronger model: a large gain over G mostly measures what the reviewer could already do.

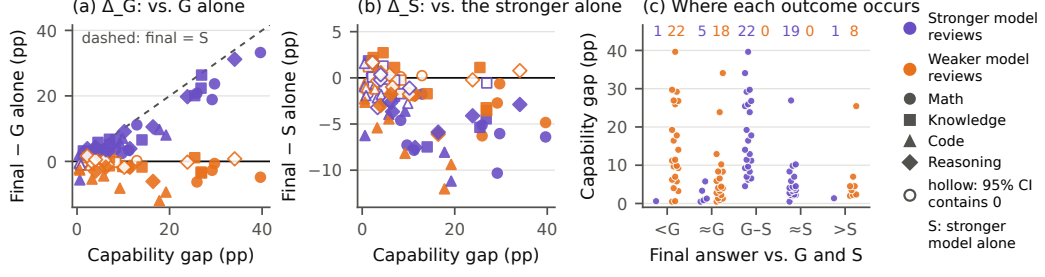


Figure 3: Each point is one review direction, coloured by who reviews whom; shape gives the domain, and hollow markers have a 95% interval containing zero. (a) Δ_G against the gap; on the dashed line the final answer matches the stronger model alone. (b) Δ_S against the gap. (c) The gap of every direction at each outcome level; numbers count the directions of each colour.

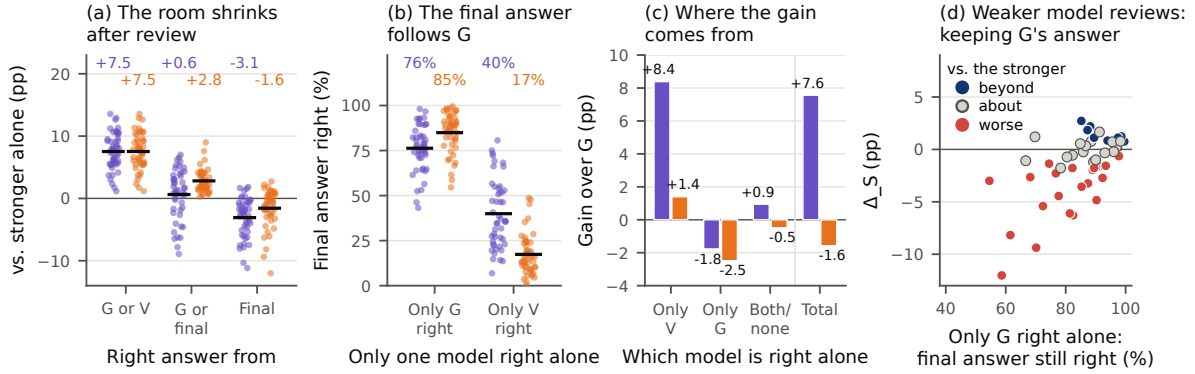


Figure 4: Why review leaves the second model's value unused. Each dot is one review direction, purple when the stronger model reviews and orange when the weaker does, as in Figure 3; black bars are means with domains weighted equally. (a) Accuracy relative to the stronger model alone of the oracle of the two solo answers (G or V right), the oracle after review (G's answer or the final answer right), and the final answer. (b) On questions only one model gets right alone, how often the final answer is right. (c) The change over G alone, by which model is right alone. (d) Weaker model reviews: on questions only G gets right alone, how often the final answer stays right, against Δ_S ; colours give the outcome level.

A weaker reviewer often makes G worse. Here G is the better model, so $\Delta_G = \Delta_S$. Twenty-two of the 48 directions end significantly below G alone, and the average change is -1.6 points $[-1.8, -1.3]$. Damage can happen at any gap but becomes the rule as the gap grows: 10 of the 13 directions with a gap of 14 points or more are worse, against 12 of the 35 closer ones (Figure 3c). In code, Llama-3.2-3B reviewing Qwen3-8B lowers Qwen3-8B's accuracy by 12.0 points.

Beating the stronger model is rare. Only 9 of 96 directions beat the stronger model alone, all in math and knowledge. Eight are a weaker model reviewing a stronger one, seven of them with a gap of at most 7 points. Being close makes a gain possible but does not deliver it: of the 26 weaker-reviews-stronger directions with a gap under 8 points, 7 beat the stronger model and 7 are significantly worse.

Outcomes track the gap, not the headroom.

Across the evaluated pairs, each 10 points of gap goes with a Δ_S lower by 1.8 points $[1.1, 3.0]$ when the stronger model reviews and by 1.5 points $[0.7, 2.5]$ when the weaker one does. One might expect the headroom to matter instead, but it is tied to the gap by construction ($h = (\delta - \gamma)/2$) and strongly negatively correlated with it within each domain ($r = -0.73 [-0.84, -0.58]$): close pairs leave the most room. With the gap held fixed, headroom has no detectable effect (-0.8 per 10 points when the stronger model reviews and $+2.3$ when the weaker one does, both intervals containing zero), while the gap keeps its slope when the stronger model reviews (2.0 $[0.8, 3.4]$) and most of it when the weaker one does (1.0 $[0.0, 2.4]$), an interval that just reaches zero). Close pairs have the most to gain from a second opinion, yet even there review usually fails to collect it.

4.2 RQ2: Why does review miss the gains a second model could offer?

Review removes the room a second model leaves. In all 48 pairs, the share of questions that at least one model answers correctly alone (the oracle) exceeds the stronger model’s accuracy, by 7.5 points on average [7.2, 7.8]. Review removes most of this room (Figure 4a): even picking, question by question, between G’s own answer and the final answer would add only 0.6 points [0.2, 1.0] over the stronger model when it reviews, and 2.8 [2.6, 2.9] when the weaker one does. In 16 of the 48 stronger-reviews-weaker directions, this oracle even falls below the stronger model alone.

The final answer stays with G. On questions where exactly one model is right alone (Figure 4b), if G was the one, the final answer stays right 76% of the time when the stronger model reviews and 85% when the weaker one does. If V was the one, the final answer is right only 40% and 17% of the time: a reviewer that solves a question alone often fails to once it has read a wrong answer, even when it is the stronger model (Figure 2). Most of these lost answers become G’s wrong one (53% and 65%). Which ones it loses depends most on the gap: the further the reviewer is above G, the more it keeps, and the further below, the fewer. Within a direction, it loses more when its own solution is long and its maximum logit low, but not when G’s answer is longer or more confident (Appendix F). A second pass does not explain this: shown its own answer in place of G’s, a reviewer keeps over 90% of these right answers. Nor does the wording of our prompt: with another prompt that has the reviewer commit first, the reviewer still kept only 55%, against 61% with ours. With thinking on, in two math directions, review still ends about 3 points below the reviewer alone, and appears to follow a stronger G but not a weaker one (Appendix F). Over all questions, the two models alone differ in correctness on 25.7% of them on average; the final answer differs from G’s on only 15.0% when the stronger model reviews and 7.2% when the weaker one does, with a drop in all 96 directions (Figure 1c; per direction in Figure 7, Appendix G). Headroom lives only where the two models differ, so once the final answer converges to G’s, it is mostly lost.

Review itself creates little. Where the two models agree, review can only fix a question both got wrong or break one both got right, and the net is

small: +0.9 points [+0.7, +1.2] when the stronger model reviews and −0.5 [−0.7, −0.3] when the weaker one does. Splitting the questions into eight cells by whether G alone, V alone and the final answer are right (Figure 4c, Table 11), a stronger reviewer’s 7.6-point gain is +8.4 on questions only V could solve, −1.8 on those only G could solve and +0.9 on the rest: almost all of it is capability transfer, not verification. For a weaker reviewer the parts are +1.4, −2.5 and −0.5.

Between close models, keeping G’s right answers decides. When a weaker model reviews a close stronger one, the directions that beat the stronger model and those that hurt it do not differ in headroom (8.7 vs. 7.4 points) or in how often the final answer takes up the reviewer’s right answers (23% vs. 25%). They differ in what happens to G’s own right answers: on questions only G gets right alone, the final answer is still right 92% of the time in the first group and 72% in the second, and across all 48 weaker-reviews-stronger directions this rate tracks Δ_S closely (Figure 4d). Beating the stronger model comes from not breaking anything, not from collecting headroom.

The gap sets how much is at stake. The gap sets how many questions only the stronger model gets right alone: 30% of all questions when the two models are 14 or more points apart, against 14% when they are closer. These are the questions on which review can gain or lose. A stronger reviewer carries its right answer into the final answer on 60% of them in far-apart pairs, against 33% in closer ones, so G gains more, yet stays below the reviewer alone. A weaker reviewer keeps G’s right answer about as often in both (87% and 84%), so it is no more persuasive in far-apart pairs; the same loss rate simply hits twice as many questions.

5 Using the second model

Cost. We count the cost of a model call with N parameters that reads t_{in} and writes t_{out} tokens as

$$c = N(t_{\text{in}} + t_{\text{out}}). \quad (4)$$

This is proportional to forward-pass compute and puts a 3B and a 72B model on one scale. A review costs $c_G + c_V$, with G’s whole answer in V’s input; the selector costs the two solo calls. Appendix D repeats every comparison with $c' = N(t_{\text{in}} + 4t_{\text{out}})$.

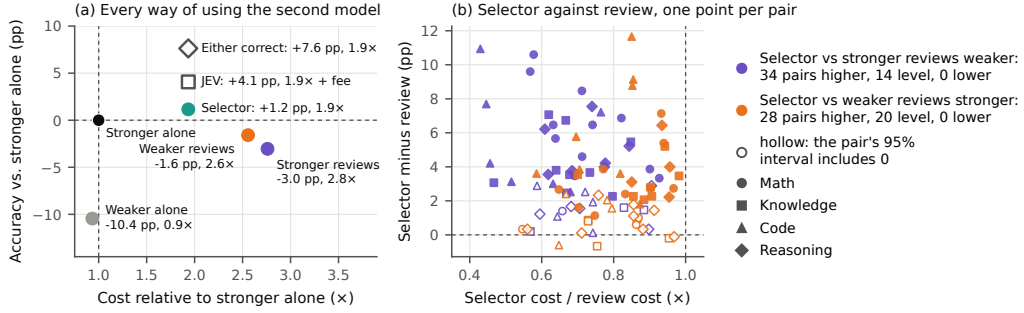


Figure 5: The cost and accuracy of using a second model; cost per question is parameters $\times$ (input + output tokens). (a) Each way of using the two models relative to the stronger model alone: means over 48 pairs with domains weighted equally, with 95% intervals; JEV also pays an API fee not on this axis. (b) The selector against each review direction, one point per pair; hollow points have a 95% interval that includes zero.

Selector. The selector lets both models answer alone and picks one answer with no further model call, so it uses the same two calls as review. A small classifier, with features and settings fixed in advance and fitted separately for each pair, estimates from numbers alone (token log-probabilities, lengths, whether the two answers agree; Appendix C) the probability that each answer is right, and the selector returns the weaker model’s answer only if its estimate is higher. It is always evaluated on questions it was not fitted on, so its math and knowledge numbers use 1,500 held-out questions and differ slightly from Section 4. Its labels come from the same development questions that are needed anyway to know which model is stronger; a rule with no labels, returning the answer with the higher mean token log-probability, is 3.7 points below the stronger model alone (“A selector without labels”, Appendix F).

A stronger reference. An external LLM judge, JEV (Li et al., 2026), reads the question and both full answers and estimates which one is right (Appendix C). It is a closed API priced in dollars, so we report it beside the selector, not against review.

5.1 RQ3: Is a second model worth its cost?

Review costs more and ends below the stronger model; selection ends above it. Figure 5a places each way by its cost and accuracy relative to the stronger model alone. Review costs 2.8 times as much for -3.0 points $[-3.5, -2.6]$ when the stronger model reviews, and 2.6 times for -1.6 $[-1.9, -1.3]$ when the weaker one does; it is significantly above the stronger model in 1 and 7 of the 48 pairs and below it in 23 and 21. The selector costs 1.9 times as much and gains 1.2 points $[0.9,$

$1.4]$, significantly above the stronger model in 19 pairs and below it in none, recovering about 15% of the 7.6 points $[7.3, 7.9]$ of room. JEV recovers more, 4.1 points $[3.8, 4.3]$, above the stronger model in 42 of 48 pairs, for an API fee of \$0.04–\$0.21 per 1,000 questions.

Review is expensive because the reviewer reads G’s whole answer, on average longer than its entire solo call, and still writes 0.8 to 1.0 times as much as alone. By domain (Table 8), the selector is above the stronger model everywhere except code $(+0.5 [-0.1, +1.0])$; review is above it in no domain.

5.2 RQ4: Select or review?

At the same two calls, choosing beats review. Pair by pair (Figure 5b), the selector is 4.2 points $[3.8, 4.6]$ more accurate than the stronger model reviewing the weaker, higher in 34 of 48 pairs and lower in none, at 0.69 times the cost. Against the weaker model reviewing the stronger, it is 2.7 points $[2.4, 3.1]$ more accurate, higher in 28 pairs and lower in none, at 0.82 times the cost. It is cheaper in every pair and more accurate in every domain. Weighting output tokens four times brings the costs close (0.89 and 0.98 times).

6 Conclusion

Gain over the generator mixes verification with capability transfer. Once the two are separated, 96 review directions in four domains tell one story: review rarely beats the stronger model alone, because a reviewer that reads the generator’s answer tends to follow it, so a second model is better used for its own answer, as a candidate to choose from, than as a reviewer. For researchers, this means reporting the stronger model alone and the decomposition

by which model is right alone (Section 3.1) alongside Δ_G , since Δ_G on its own credits a stronger reviewer’s own ability to review. For practitioners, it means checking, before adding a review step, whether a single call to the better model already wins; if a second call is worth paying for, choosing between the two models’ answers is the baseline to beat. Asking the reviewer to commit to its own answer first did not stop it from following G.

Limitations

In this paper, we showed that the standard evaluation of review confuses verification with capability transfer, a distinction that is essential for deciding whether a second model call is worth its cost. Some limits of scope remain.

Other models. We study open-weight models from 3B to 72B parameters and one reviewer with thinking on in two math directions. Testing closed frontier models and more thinking-on models is left for future work.

Other tasks. All our tasks are in English and graded by program. Extending the comparison to open-ended generation, where no program can grade an answer, is left for future research.

Why the reviewer follows G. We measured how often the final answer follows G and what this costs a pair, and checked one alternative prompt on four directions (in math, the reviewer often skipped committing to its own answer first), but not what in the reviewer causes it. Other prompts, several rounds and training are left for future work. Likewise, verification and capability transfer are defined by outcome, not by what happens inside the reviewer: a right answer the reviewer also gives alone counts as transfer, even if it was reached by checking G’s.

References

- Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam Mausam. 2024. [Automix: Automatically mixing language models](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 131000–131034. Curran Associates, Inc.
- ByteDance Seed, Yuyu Zhang, Jing Su, Yifan Sun, Chenguang Xi, Xia Xiao, Shen Zheng, Anxiang Zhang, Kaibo Liu, Daoguang Zan, Tao Sun, Jinhua Zhu, Shulin Xin, Dong Huang, Yetao Bai, Lixin Dong, Chao Li, Jianchong Chen, Hanzhi Zhou, and 8 others. 2025. [Seed-coder: Let the code model curate data for itself](#). *Preprint*, arXiv:2506.03524.
- Lingjiao Chen, Matei Zaharia, and James Zou. 2024. [FrugalGPT: How to use large language models while reducing cost and improving performance](#). *Transactions on Machine Learning Research*. Featured Certification.
- Hyeon Kyu Choi, Jerry Zhu, and Sharon Li. 2026. [When identity skews debate: Anonymization for bias-reduced multi-agent reasoning](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14284–14311, San Diego, California, United States. Association for Computational Linguistics.
- Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024. [Hybrid LLM: Cost-efficient and quality-aware query routing](#). In *The Twelfth International Conference on Learning Representations*.
- Idil Gozel. 2026. [Commit-first llm judging inherits the judge’s own errors](#). *Preprint*, arXiv:2609.00088.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenyong Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. [FOLIO: Natural language reasoning with first-order logic](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the math dataset](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.
- Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. [Large language models cannot self-correct reasoning yet](#). In *The Twelfth International Conference on Learning Representations*.
- Binyuan Hui, Jian Yang, Zeyu Cui, Jiayi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, and 5 others. 2024. [Qwen2.5-coder technical report](#). *Preprint*, arXiv:2409.12186.

- Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. [LLM-blender: Ensembling large language models with pairwise ranking and generative fusion](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, Toronto, Canada. Association for Computational Linguistics.
- Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. [When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs](#). *Transactions of the Association for Computational Linguistics*, 12:1417–1440.
- Vira Kasprova, Amruta Parulekar, Abdulrahman AlRabah, Krishna Agaram, Ritwik Garg, Sagar Jha, Nimet Beyza Bozdog, and Dilek Hakkani-Tür. 2026. [Too polite to disagree: Understanding sycophancy propagation in multi-agent systems](#). In *Proceedings of the 27th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 795–814, Atlanta, Georgia, USA. Association for Computational Linguistics.
- Yubo Li, Yidi Miao, Ramayya Krishnan, and Rema Padman. 2026. [Jev-as-a-judge: Accept when confident, escalate when unsure](#). *Preprint*, arXiv:2609.26550.
- Shalev Lifshitz, Sheila A. McIlraith, and Yilun Du. 2025. [Multi-agent verification: Scaling test-time compute with multiple verifiers](#). In *Second Conference on Language Modeling*.
- Ruihan Lin and Jiheng Zhang. 2026. [When to screen, when to bypass: Llm-judges in resource-scarce ai-human workflow](#). *Preprint*, arXiv:2603.13870.
- Jack Lu, Ryan Teehan, Jinran Jin, and Mengye Ren. 2026. [When does verification pay off? a closer look at llms as solution verifiers](#). *Preprint*, arXiv:2512.02304.
- Sasha Luccioni, Yacine Jernite, and Emma Strubell. 2024. [Power hungry processing: Watts driving the cost of ai deployment?](#) In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency, FAccT ’24*, page 85–99, New York, NY, USA. Association for Computing Machinery.
- Meta AI. 2024. [Llama 3.2: Revolutionizing edge ai and vision with open, customizable models](#). Meta AI.
- Mistral AI. 2024. [Un ministral, des ministraux](#). Mistral AI.
- Mistral AI. 2025. [Mistral Small 3](#). Mistral AI.
- Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2025. [RouteLLM: Learning to route LLMs from preference data](#). In *The Thirteenth International Conference on Learning Representations*.
- Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. [Llm evaluators recognize and favor their own generations](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 68772–68802. Curran Associates, Inc.
- Jiaming Qu, Lucheng Fu, and Yibo Hu. 2026. [Easier to mislead than to correct: Harmful and beneficial revision in llm conformity](#). *Preprint*, arXiv:2606.01637.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. [GPQA: A graduate-level google-proof q&a benchmark](#). In *First Conference on Language Modeling*.
- Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Esin DURMUS, Zac Hatfield-Dodds, Scott R Johnston, Shauna M Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. 2024. [Towards understanding sycophancy in language models](#). In *The Twelfth International Conference on Learning Representations*.
- Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. 2024. [Large language model routing with benchmark datasets](#). In *First Conference on Language Modeling*.
- Nishad Singhi, Hritik Bansal, Arian Hosseini, Aditya Grover, Kai-Wei Chang, Marcus Rohrbach, and Anna Rohrbach. 2025. [When to solve, when to verify: Compute-optimal problem solving and generative verification for LLM reasoning](#). In *Second Conference on Language Modeling*.
- Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024a. [Large language models are not fair evaluators](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. 2024b. [Mmlu-pro: A more robust and challenging multi-task language understanding benchmark](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 95266–95290. Curran Associates, Inc.
- Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khoshabi, and Yejin Choi. 2023. [Generating sequences by learning to self-correct](#). In *The Eleventh International Conference on Learning Representations*.

Zhiyuan Weng, Guikun Chen, and Wenguan Wang. 2025. [Do as we do, not as you think: the conformity of large language models](#). In *The Thirteenth International Conference on Learning Representations*.

Zuodong Xiang, Yike Zhang, YueMing Zhang, and Hailu Xu. 2026. [Cross-model llm code review: Should you use claude to review codex or vice versa?](#) *Preprint*, arXiv:2607.21656.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025b. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.

Chih-Hsuan Yang, Jingyan Jiang, Vikram Vasudevan, Cheng-Hau Yang, Huihuo Zheng, Le Chen, Eliu A. Huerta, Venkatram Vishwanath, Ian T. Foster, and Rajeev Thakur. 2026. [Precise but uncoupled: Reviewer precision does not guarantee critique uptake in multi-agent math reasoning](#). *Preprint*, arXiv:2607.15388.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. [Reclor: A reading comprehension dataset requiring logical reasoning](#). In *International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. 2023. [Judging LLM-as-a-judge with MT-bench and chatbot arena](#). In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Chenyu Zhou. 2026. [More convincing, not more correct: Self-play reward hacking of reference-free llm judges](#). *Preprint*, arXiv:2607.05904.

Yefan Zhou, Austin Xu, Yilun Zhou, Janvijay Singh, Jiang Gui, and Shafiq Joty. 2026. [Variation in verification: Understanding verification dynamics in large language models](#). In *The Fourteenth International Conference on Learning Representations*.

Xiaochen Zhu, Caiqi Zhang, Tom Stafford, Nigel Collier, and Andreas Vlachos. 2025. [Conformity in large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3854–3872, Vienna, Austria. Association for Computational Linguistics.

Terry Yue Zhuo, Vu Minh Chien, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon

Brunner, Chen GONG, James Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kadour, Ming Xu, Zhihan Zhang, and 14 others. 2025. [Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions](#). In *The Thirteenth International Conference on Learning Representations*.

Code and per-question results are available at <https://github.com/Kamilaabb/verification-or-capability-transfer>.

A Model pairs

Within each domain, we sort the models by their accuracy on the development questions (for code and reasoning, all questions) into strong, middle and weak tiers. For a pair, we count the development questions that only the weaker model answers correctly, which is the pair’s headroom on those questions, and in each of the six tier combinations we take the pair with the most and the pair with the fewest. Table 4 lists the 48 pairs.

A few pairs needed a further rule, fixed before any test-set review results; Table 3 lists them. Where the chosen pair included Qwen2.5-72B, we took the next pair in headroom order to save compute, as reviewing with it takes two H100 GPUs. We later ran the four original pairs in both directions as well; Appendix E includes them in the analysis over all completed pairs.

B Prompts

Every call sends a system message and a user message. Each domain has one solve prompt and one review prompt, and reasoning uses the knowledge prompts unchanged. Both models of a pair answer alone with the same solve prompt. In the templates below, `{{problem}}`, `{{question}}` and `{{options}}` are replaced by the question, and `{{y0}}` by G’s complete response as plain text. Options are listed one per line as “A. option”. A math problem is the MATH problem statement and a code task is the BigCodeBench-Instruct prompt. A ReClor question is the passage, a blank line and the question. A FOLIO question gives the premises and the conclusion and asks “Based only on the premises, is the conclusion true, false, or uncertain?”, with the options True, False and Uncertain. GPQA lists the correct answer first, so each question’s options are shuffled with a fixed permutation.

Math, solve

System:

Cell	Case	Pair used
<i>Ties and small tiers</i>		
Knowledge, strong-middle, most	Two pairs tie	The pair already run on the development questions
Knowledge, strong-weak, fewest	Two pairs tie	The pair whose weaker model is weaker
Reasoning, middle-middle	The middle tier has two models	Its one pair
Reasoning, strong-middle, fewest	Two pairs tie	Both (12 pairs in all)
Knowledge, strong-strong, fewest	Only three strong models; without Qwen2.5-72B one pair is left	Qwen2.5-72B with Mistral-Small-24B, kept
<i>Qwen2.5-72B replaced by the next pair</i>		
Math, strong-strong, most	Qwen2.5-72B with Qwen3-4B (79)	Qwen3-4B with Qwen3-8B (70)
Math, strong-strong, fewest	Qwen2.5-72B with Qwen3-32B (35)	Qwen3-8B with Qwen3-32B (39)
Knowledge, strong-middle, fewest	Qwen2.5-72B with Qwen2.5-7B (48)	Mistral-Small-24B with Qwen2.5-7B (64)
Knowledge, strong-weak, fewest	Qwen2.5-72B with Llama-3.2-3B (35)	Mistral-Small-24B with Llama-3.2-3B (44)

Table 3: Pairs chosen by a further rule. Numbers in parentheses: development questions that only the weaker model answers correctly.

Solve the mathematical problem provided by the user. Use mathematically sound reasoning and calculations to produce a complete solution.
End with exactly one final answer in LaTeX boxed form: `\boxed{...}`.

User template:
Problem:
{{problem}}

Math, review

System:
Review the proposed mathematical solution for correctness.
The proposed solution may be correct or incorrect. Carefully check the reasoning, calculations, assumptions, and final answer.
If the proposed solution is correct, retain it.
If it contains any mathematical error, correct the error and produce a correct solution.
Return a complete final solution to the original problem. End with exactly one final answer in LaTeX boxed form: `\boxed{...}`.

User template:

Problem:
{{problem}}

Proposed solution:
{{y0}}

Knowledge and reasoning, solve

System:
Answer the multiple-choice question provided by the user. Think step by step before answering.
The last line of your response must be of the form "Answer: X", where X is the letter of exactly one option.

User template:
Question:
{{question}}

Options:
{{options}}

Knowledge and reasoning, review

System:
Review the proposed solution to the multiple-choice question for correctness.
The proposed solution may be correct or incorrect. Carefully check its facts, its reasoning, and its final choice.
If the proposed solution is correct, retain its answer. If it contains any error, correct the error and produce a correct solution.
Return a complete final solution to the original question. The last line of your response must be of the form "Answer: X", where X is the letter of exactly one option.

User template:
Question:
{{question}}

Options:
{{options}}

Proposed solution:
{{y0}}

Code, solve

System:
Complete the Python programming task provided by the user. Use sound reasoning to produce a correct and complete solution.
End with exactly one complete, self-contained Python code block (````python ... ````) that includes all required imports and the requested function definition.

User template:
Task:
{{problem}}

Code, review

System:
Review the proposed solution to the Python

programming task for correctness.
 The proposed solution may be correct or incorrect. Carefully check its logic, edge cases, use of library APIs, and whether it meets every requirement of the task.
 If the proposed solution is correct, retain it.
 If it contains any error, correct the error and produce a correct solution.
 Return a complete final solution to the original task. End with exactly one complete, self-contained Python code block (````python ... ````) that includes all required imports and the requested function definition.

User template:
 Task:
 {{problem}}

Proposed solution:
 {{y0}}

Math, commit-first review (Appendix F only)

System:
 Solve the mathematical problem provided by the user. A proposed solution is also given, but do not use it until you have committed to your own answer.
 First, solve the problem independently and state your answer on its own line in the form "My answer: ..." (without `\boxed`). Only then read the proposed solution and compare it with your own. Either one may be correct or incorrect: carefully check the reasoning, calculations, assumptions, and final answer of both, and decide which final answer is correct.
 Return a complete final solution to the original problem. Do not use `\boxed` before the end. End with exactly one final answer in LaTeX boxed form: `\boxed{...}`.

User template: as for math review.

Knowledge, commit-first review (Appendix F only)

System:
 Answer the multiple-choice question provided by the user. A proposed solution is also given, but do not use it until you have committed to your own answer.
 First, think step by step and answer the question independently, and state your answer on its own line in the form "My answer: X", where X is the letter of one option. Only then read the proposed solution and compare it with your own. Either one may be correct or incorrect: carefully check the facts, the reasoning, and the final choice of both, and decide which option is correct.
 Return a complete final solution to the original question. The last line of your response must be of the form "Answer: X", where X is the letter of exactly one option.

User template: as for knowledge review.

C Selector and JEV

The selector sees, for each output, token log-probabilities and entropy, length, and whether an answer can be parsed; for the pair, whether the two answers agree; and for the question, its length and, where the dataset provides one, its subject. Its probability estimate is the average of an L2 logistic regression and a small gradient-boosted tree. Math and knowledge use their 1,500 held-out test questions; code and reasoning have fewer questions and use five-fold cross-fitting. All numbers in Section 5 use these questions, so for math and knowledge they differ slightly from Section 4, which uses all 2,500.

JEV reads the question and both full answers and returns the probability that each answer is right. A judge can favour an answer for its position (Wang et al., 2024a), so we call it twice, once with each answer first, and average the two. Where the two answers agree there is nothing to pick and no call is made. If the call fails or the two averaged probabilities are equal, we return the stronger model’s answer, as the selector does. Every response reports the model version jev-1.13.0.

Why we do not add a third call. A reviewer could also answer the question alone first and then, in a further call, read its own answer and G’s and choose between them. This design picks between two independent answers, as our selector does, but adds a third LLM call. We compare ways of using a second model at the same number of calls, which Huang et al. (2024) argue is crucial for methods that use several model responses, so we leave it out. Its result would also be hard to read, because the last call can be anchored to either answer: committing to an answer first does not remove the anchor but moves it from the candidate to the judge’s own answer (Gozel, 2026), and a model that has answered first is still misled by its peers (Qu et al., 2026). Writing its own answer before reading G’s within one call (Zhou, 2026) keeps the number of calls at two, so we ran this variant on four directions: the reviewer still kept only 55% of its right answers on questions only it got right alone, against 61% with our prompt (Appendix F). JEV is a third call of a different kind: it writes no text, only label probabilities, at a small fraction of a reasoning judge’s fee and latency (Li et al., 2026). In our runs it cost \$0.04–\$0.21 per 1,000 questions, and a sample of 196 calls took a median of 0.28 seconds. We report it only as a reference, not as a way of

using the second model.

Selector details. The selector estimates, for each of the two answers, the probability that it is correct and returns the answer with the higher estimate; a tie goes to the stronger model. Each estimate averages two models over five random seeds: an L2 logistic regression on that answer’s own features and the question features, and a gradient-boosted tree on all features of Table 5, including the other answer and the pair. Hyperparameters are chosen by five-fold cross-validation on the training questions with log loss (logistic regression: $C \in \{0.01, 0.1, 1, 10\}$; tree: depth 2 or 3, learning rate 0.05 or 0.1, 100 or 300 iterations). Missing values are replaced by the median plus an indicator for the logistic regression and left to the tree. Each pair has its own selector. In math and knowledge it is trained on that pair’s 1,000 development questions; in code and reasoning, cross-fitting nests this search inside each of the five outer folds. Every feature is computed from the question and the two outputs with their token probabilities; the reference answer, the grades and the question identifiers never enter. Code has the most features because a program’s structure can be read without running it. Reasoning reuses the knowledge selector unchanged, so its 18 subject and source indicators, defined for MMLU-Pro, are always zero there.

D Cost with output tokens weighted four times

Section 5 counts input and output tokens alike. Generating a token usually costs more than reading one, so Table 6 repeats the cost ratios with $c' = N(t_{\text{in}} + 4t_{\text{out}})$. Accuracy does not depend on the cost measure and is unchanged. Review becomes cheaper relative to the stronger model alone, 2.2 and 2.0 times instead of 2.8 and 2.6, since much of a reviewer’s cost is reading G’s whole answer, while the selector’s cost hardly moves (1.9 times). Against review, the selector then costs 0.89 times as much as the stronger model reviewing and 0.98 times as much as the weaker one reviewing, and it is cheaper in 34 and 28 of the 48 pairs instead of all 48; against the weaker model reviewing, it costs slightly more in math, knowledge and reasoning. With output weighted this heavily, choosing between two answers keeps its accuracy advantage over review and still costs less on average.

E Per-domain results and robustness

Bootstrap. With the models and pairs fixed by design, the main intervals measure the uncertainty from which questions were sampled. For one direction, we resample questions with replacement, keeping each question’s paired outcomes together (2,000 replicates, percentile 95% intervals), within the development and test parts and, in reasoning, within each source dataset. In each replicate, all directions of a domain share one resample of questions. The regressions on the gap are fitted separately for the two kinds of direction, and since they compare pairs, their bootstrap also resamples pairs within each domain.

Per domain. Table 7 splits the averages and outcome levels of Section 4 by domain. Review falls below the stronger model on average in every domain and in both kinds of direction, with one exception: a weaker model reviewing a stronger one in knowledge is level with it (+0.1 [−0.3, +0.5]), and five of the eight weaker-reviewer directions beyond the stronger model are there. Code is where review does worst: a stronger reviewer lifts G by only 2.7 points, and no weaker reviewer improves on G. Table 8 gives the comparisons of Section 5 by domain: the selector is above the stronger model in every domain except code, and above both review directions in every domain.

Leaving out one model. Models recur across pairs, so one model could drive an average. We repeat each average after removing every direction that involves one of the 12 models, and Table 9 gives the range over the 12 repetitions. No average changes sign.

All completed pairs. Besides the 48 pairs, we ran review on further pairs during the study, some in one direction only, including the four original pairs with Qwen2.5-72B (Appendix A): 131 review directions in all. These never enter the main analysis. Pooled over both kinds of direction, the averages over all of them stay close to those over the 48 pairs (Table 9).

Resampling model pairs. The intervals in the main text keep the evaluated pairs fixed and resample questions. Table 10 also resamples model pairs within each domain, keeping the two directions of a pair as one unit, along with the questions. The intervals widen, since pairs differ from one another, but every average keeps its sign. The one

that then reaches below zero is the oracle after the stronger model reviews, which Section 4.2 already describes as almost no room left. Models recur across pairs, so pairs are not fully independent and this is a robustness check rather than an interval over a population of model pairs.

F Further checks

All eight cells. Table 11 splits the questions by whether G alone, V alone and the final answer are right. Two rows exist only because of review: questions both models get wrong that review gets right, and questions both get right that review gets wrong. A choice between the two independent answers, such as the selector of Section 5, can produce neither. When the stronger model reviews, these rows hold 2.9% and 1.9% of the questions, while the reviewer’s own right answers that review takes up hold 8.4%, and those it loses 9.8%.

Does the reviewer follow G? When the stronger model reviews, its final answer is the same as G’s on 57.7% [57.2, 58.2] of the questions of Section 4, against 41.8% [41.2, 42.3] for its own solo answer; when the weaker model reviews, the final answer is G’s on 69.1% [68.7, 69.4]. Answers count as the same when the extracted answers are identical (the option letter in knowledge and reasoning, the whole final program in code). Of the right answers a reviewer loses on questions only it got right alone, most end on G’s wrong answer: 53.3% [52.2, 54.5] when the stronger model reviews and 65.0% [63.7, 66.3] when the weaker one does. In code, the reviewer’s own program never equals G’s, yet its final program does on 21% of the questions.

When does the reviewer keep its own answer?

On the 21,247 questions where G was wrong and V right alone, we relate whether the final answer stays right to the reviewer’s strength relative to G and to features of the two solo answers, compared within each direction (logistic regression with direction fixed effects; Table 12). Across directions, the further a stronger reviewer is above G, the more of these answers it keeps (+13.8 points per 10 points of gap, $r = 0.74$), and the further a weaker reviewer is below G, the fewer (−4.6 points, $r = -0.41$). Within a direction, the reviewer keeps fewer of them the longer its own solution and more the higher its mean maximum logit. G’s answer shows no association: the intervals for its length, maximum logit and log-probability all contain 1,

except that a weaker reviewer keeps more when G’s output was cut off at the token limit. This matches the thinking check below, where the reviewer stops following a weaker G but still follows a stronger one. These are correlations, not causal effects: we vary none of these features, and the questions are selected because the reviewer was right alone.

Self-review as a control. Without external feedback, models do not improve their own reasoning and can even lose accuracy (Huang et al., 2024). This makes self-review a control for following G: the same reviewer, prompt and questions, with the reviewer’s own solo answer shown in place of G’s. If the review step itself made reviewers drop right answers, they would drop them under self-review too. On 1,000 MATH questions used during development, with the same prompt, Qwen3-8B reviewing its own answers kept 253 of its 254 right answers that Ministral-8B had missed, but kept 157 when it reviewed Ministral-8B; it got 783 questions right alone, 788 under self-review and 743 after reviewing Ministral-8B. Ministral-8B kept all 22 of its right answers that Qwen3-8B had missed under self-review and none after reviewing Qwen3-8B, and 20 of the 22 it lost became Qwen3-8B’s answer. On the first 500 held-out questions of math and knowledge, we ran self-review for the four models that review in the design directions (11 directions; two more take the reviewer’s solo answers from a different run and are left out). Summed over the directions, of the right answers only the reviewer had, self-review kept 467 of 468 in math and 449 of 493 in knowledge, while reviewing G kept 306 and 218. Each model’s accuracy under self-review stayed within five questions of its accuracy alone (414 and 416, 407 and 405 in math; 318 and 313, 281 and 283 in knowledge). Self-review changes the prompt and the context just as reviewing G does, and its questions are selected the same way, yet it keeps 99.8% of these right answers in math and 91% in knowledge, against 65% and 44% after reviewing G. So the new prompt and context, and regression to the mean on questions chosen because the reviewer got them right, explain at most a small part of the losses after reviewing G. The rest comes from anchoring on G’s answer, and most of the lost answers become G’s answer (see above). Self-review may also keep answers because models prefer their own outputs (Panickssery et al., 2024); either way, the reviewer keeps the answer it is shown, and the losses come from being shown

G’s.

A prompt without the instruction to retain.

Our review prompt tells the reviewer to retain a proposed solution that is correct, which makes G’s answer the default, so the reviewer might follow G because the prompt tells it to. We reran four directions on the same 500 held-out questions as the self-review check with only the review prompt changed: the reviewer first solves the question and states its own answer on a line of its own, then reads G’s solution, told that either may be right or wrong, and returns a final solution (Appendix B). The instruction to retain is gone. This is the one-call commit-first protocol of Zhou (2026). The directions are Qwen3-8B reviewing Llama-3.1-8B and Qwen3-32B in math, and Qwen3-8B reviewing Llama-3.2-3B and Qwen2.5-7B reviewing Qwen3-4B in knowledge, two where the stronger model reviews and two where the weaker one does. Of the right answers only the reviewer had, the new prompt kept 231 of 421 (55%), against 256 (61%) with our prompt and 404 (96%) under self-review; per direction, 143, 4, 69 and 15, against 157, 3, 79 and 17. The change over the stronger model alone moved from -6.8 , $+0.6$, -7.8 and $+2.0$ points to -9.2 , $+1.0$, -5.4 and $+2.2$. Whenever the committed answer was right and G’s was wrong, the final answer stayed right (143 of 144 questions). The answers are lost earlier: with G’s solution already in its input, the reviewer often commits to an answer that is not its own solo answer, which in knowledge it gives on only 64% and 66% of the questions with a committed answer (in math the reviewer wrote a committed answer on only 211 and 310 of the 500, so there the variant was often not carried out as written). Of the committed answers that differ from the reviewer’s solo answer, 270 of 479 are G’s answer. Following G is not an effect of the instruction to retain.

Thinking on. All our runs keep Qwen3’s thinking off. A reviewer that reasons at length before answering might check G’s solution instead of following it. We reran Qwen3-8B with thinking on, alone and reviewing Llama-3.1-8B and Qwen3-32B, on the same 500 held-out math questions as the prompt check, with G’s answers unchanged. Qwen advises against greedy decoding with thinking on, so we sample with its recommended settings (temperature 0.6, top-p 0.95, top-k 20) and allow up to 32,768 output tokens. Thinking raises Qwen3-8B alone from 414 to 482 right answers,

and review still falls below it: by 2.8 points [1.0, 4.6] reviewing Llama-3.1-8B and 3.0 [0.8, 5.2] reviewing Qwen3-32B, mostly on questions only the reviewer got right alone. How it loses them depends on G. Reviewing the weaker Llama-3.1-8B, it keeps 252 of 270 of these answers (93%, against 77% with thinking off), and only 3 of the 18 it loses become Llama-3.1-8B’s answer. Reviewing the stronger Qwen3-32B, it now answers many more of these questions alone (64 against 23), but still loses about as many (21 against 20), and 14 of them become Qwen3-32B’s answer (17 with thinking off). Because we sample with thinking on, a question answered right once alone is not always answered right again; this can explain losses to other answers, but not losses to G’s. So with thinking on, the reviewer appears to stop following a weaker G but still follows a stronger one; why, and whether this holds for other models and domains, is left for future work. Each review output grows from about 800 tokens to 2,900 and 4,000.

Output length. Few outputs stop at the 4,096-token limit: 1.7% of review outputs when the stronger model reviews and 1.8% when the weaker one does, against 1.8% for the stronger model alone and 3.3% for the weaker one. Math has the most (4.4–4.5% of review outputs, 5.0–7.4% of solo answers). Review outputs are cut off no more often than the stronger model’s own answers, so the limit is unlikely to explain why review loses right answers.

A selector without labels. Our selector is trained on development questions with known answers. We also tried a rule that needs no answers: return the answer whose output has the higher mean token log-probability, and the stronger model’s answer on ties. It is 3.7 points [3.4, 4.1] below the stronger model alone, below it in every domain, and significantly below it in 24 of the 48 pairs and above it in none. In practice, however, knowing which model is stronger on one’s own task also requires setting aside part of the data with known answers and testing both models on it. Review needs this to decide which model reviews, and using the stronger model alone needs it to decide which model to use. The selector uses the same development questions and nothing more, so on this point the comparison is fair. Under this fair comparison, the selector is 1.2 points [0.9, 1.4] above the stronger model alone, and 4.2 [3.8, 4.6] and 2.7 [2.4, 3.1] points above review by the stronger and by the weaker model

(Table 8).

Gap and headroom on held-out questions only.

In math and knowledge, the regression of Section 4 uses both development and held-out questions. On the held-out questions alone it is unchanged: per 10 points of gap, Δ_S is lower by 1.9 [1.1, 3.2] when the stronger model reviews and by 1.5 [0.7, 2.5] when the weaker one does; with headroom held fixed, by 1.9 [0.6, 3.5] and 1.1 [0.0, 2.3], while headroom has no detectable effect (-0.1 [-3.7 , $+4.6$] and $+2.2$ [-1.4 , $+5.7$]); and the correlation between gap and headroom is $r = -0.73$ [-0.84 , -0.58]. Since $h = (\delta - \gamma)/2$, a larger gap leaves less headroom at a given disagreement rate δ ; in our pairs, δ does not grow with the gap fast enough to offset this.

G Additional figures

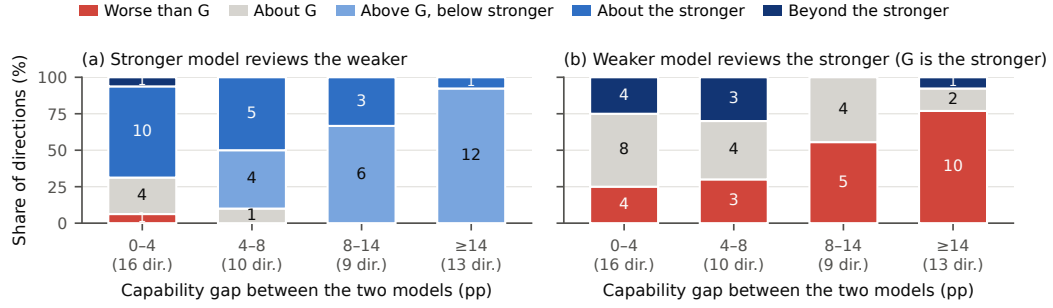


Figure 6: Where review lands, by who reviews whom and the gap. Each bar is one gap range, split by outcome level (paired 95% intervals); numbers count directions. (a) The stronger model reviews the weaker. (b) The weaker reviews the stronger; G is then the stronger model, so only three levels occur.

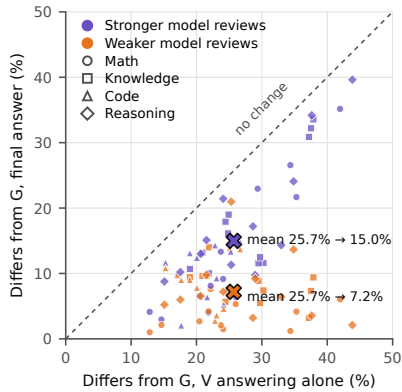


Figure 7: After review, the final answer agrees with G more often. Each point is one review direction. x: how often the reviewer answering alone differs from G in correctness; y: how often the final answer does. Points below the dashed line moved toward G; crosses are means with domains weighted equally.

Domain	Tiers	Headroom	Stronger	Acc.	Weaker	Acc.	Gap	h
Math	strong-strong	most	Qwen3-8B	80.3	Qwen3-4B	78.3	2.0	6.3
	strong-strong	fewest	Qwen3-32B	85.1	Qwen3-8B	80.6	4.5	4.2
	strong-middle	most	Qwen3-4B	78.3	Mistral-Small-24B	65.4	12.9	5.4
	strong-middle	fewest	Qwen3-32B	85.1	Qwen2.5-3B	59.2	25.9	1.7
	strong-weak	most	Qwen3-8B	80.6	Ministral-8B	50.8	29.7	2.3
	strong-weak	fewest	Qwen3-8B	80.6	Llama-3.1-8B	40.9	39.6	1.2
	middle-middle	most	Qwen2.5-7B	68.8	Mistral-Small-24B	65.4	3.5	8.5
	middle-middle	fewest	Qwen2.5-7B	68.8	Qwen2.5-3B	59.2	9.7	6.1
	middle-weak	most	Qwen2.5-3B	59.2	Ministral-8B	50.8	8.3	7.9
	middle-weak	fewest	Qwen2.5-7B	68.8	Llama-3.2-3B	39.6	29.2	3.0
	weak-weak	most	Llama-3.1-8B	40.9	Llama-3.2-3B	39.6	1.3	10.4
	weak-weak	fewest	Ministral-8B	50.8	Llama-3.2-3B	39.6	11.2	7.4
Knowledge	strong-strong	most	Qwen3-32B	69.8	Mistral-Small-24B	65.2	4.5	8.2
	strong-strong	fewest	Qwen2.5-72B	72.2	Mistral-Small-24B	65.2	7.0	6.0
	strong-middle	most	Mistral-Small-24B	65.2	Qwen3-8B	63.9	1.4	10.3
	strong-middle	fewest	Mistral-Small-24B	65.2	Qwen2.5-7B	56.9	8.4	8.0
	strong-weak	most	Qwen3-32B	69.8	Qwen2.5-3B	42.8	26.9	5.5
	strong-weak	fewest	Mistral-Small-24B	65.2	Llama-3.2-3B	38.4	26.8	5.4
	middle-middle	most	Qwen3-4B	59.1	Qwen2.5-7B	56.9	2.2	11.3
	middle-middle	fewest	Qwen3-8B	63.9	Qwen2.5-7B	56.9	7.0	9.0
	middle-weak	most	Qwen2.5-7B	56.9	Qwen2.5-3B	42.8	14.0	7.8
	middle-weak	fewest	Qwen3-8B	63.9	Llama-3.2-3B	38.4	25.4	5.9
	weak-weak	most	Qwen2.5-3B	42.8	Ministral-8B	40.5	2.4	13.6
	weak-weak	fewest	Qwen2.5-3B	42.8	Llama-3.2-3B	38.4	4.4	12.9
Code	strong-strong	most	Qwen3-32B	43.8	Mistral-Small-24B	43.3	0.5	8.3
	strong-strong	fewest	Seed-Coder-8B	43.9	Mistral-Small-24B	43.3	0.6	7.3
	strong-middle	most	Mistral-Small-24B	43.3	Qwen3-8B	42.4	0.8	9.6
	strong-middle	fewest	Mistral-Small-24B	43.3	Ministral-8B	37.5	5.8	6.6
	strong-weak	most	Qwen3-32B	43.8	Qwen2.5-7B	33.9	9.9	6.5
	strong-weak	fewest	Seed-Coder-8B	43.9	Llama-3.2-3B	24.6	19.2	3.7
	middle-middle	most	Qwen3-4B	38.0	Ministral-8B	37.5	0.5	10.6
	middle-middle	fewest	Qwen2.5-Coder-7B	40.7	Ministral-8B	37.5	3.2	7.2
	middle-weak	most	Qwen3-4B	38.0	Qwen2.5-7B	33.9	4.1	9.5
	middle-weak	fewest	Qwen3-8B	42.4	Llama-3.2-3B	24.6	17.8	3.7
	weak-weak	most	Qwen2.5-7B	33.9	Qwen2.5-3B	31.1	2.8	10.7
	weak-weak	fewest	Qwen2.5-7B	33.9	Llama-3.2-3B	24.6	9.3	7.9
Reasoning	strong-strong	most	Qwen3-8B	73.8	Mistral-Small-24B	71.6	2.2	9.2
	strong-strong	fewest	Qwen3-32B	77.8	Qwen3-8B	73.8	4.0	5.5
	strong-middle	most	Mistral-Small-24B	71.6	Qwen3-4B	67.6	4.0	10.7
	strong-middle	fewest	Qwen3-8B	73.8	Qwen3-4B	67.6	6.2	5.7
	strong-middle	fewest	Qwen3-32B	77.8	Qwen3-4B	67.6	10.2	5.7
	strong-weak	most	Mistral-Small-24B	71.6	Qwen2.5-3B	55.2	16.4	9.2
	strong-weak	fewest	Qwen3-32B	77.8	Llama-3.2-3B	43.7	34.1	4.9
	middle-middle	only	Qwen3-4B	67.6	Qwen2.5-7B	61.7	5.9	9.1
	middle-weak	most	Qwen2.5-7B	61.7	Qwen2.5-3B	55.2	6.5	11.2
	middle-weak	fewest	Qwen3-4B	67.6	Llama-3.2-3B	43.7	23.9	6.9
	weak-weak	most	Qwen2.5-3B	55.2	Ministral-8B	51.8	3.3	12.7
	weak-weak	fewest	Qwen2.5-3B	55.2	Llama-3.2-3B	43.7	11.4	10.8

Table 4: The 48 model pairs. Tiers and the headroom end (most or fewest questions that only the weaker model answers correctly) come from the development questions; accuracy, gap and headroom h (%) are over all questions used in the analyses (2,500 in math and knowledge, 832 in code, 901 in reasoning). In math, the Qwen3-4B and Qwen3-8B pair uses their answers from an earlier run on the test questions, so Qwen3-8B’s accuracy differs by 0.2 points from its other pairs.

	Math	Knowledge and reasoning	Code
<i>Each answer</i>			
Basic	7: mean log-probability and largest logit over the output and over math expressions; answer parseable; question and output length in characters	22: answer parseable; number of options; question length; prompt and output tokens; truncated; 12 token statistics over the output; log-probability, entropy, margin and largest logit at the answer letter	17: question length; prompt and output tokens; truncated; number of code blocks; 12 token statistics over the output
Token spans	33: 11 token statistics each over the output, the math expressions and the boxed answer	–	22: 11 token statistics each over the code and the prose
Content	4: output tokens; truncated; number of boxed answers; self-correction words	16: spread of the answer-letter probability over the options (7); letters mentioned, answer lines, whether the first letter is the last, hedging, elimination and self-correction words (6); digits, numbers and equals signs (3)	26: program structure from the syntax tree (16); use of the required libraries and of the exceptions the task asks for (3); undefined names, unseeded randomness, prints, main guard, file access (5); output characters and self-correction words (2)
Total	44	38	65
Question	8: difficulty level; 7 subject indicators	18: 14 subject and 4 source indicators (zero in reasoning)	3: required libraries; whether and how many exceptions the task asks for
Pair	19: answers equivalent; overlap of math expressions; same answer type; parseable and truncated in one answer only (2); differences of 14 answer features	21: same letter; both parseable; parseable and truncated in one answer only (2); divergence of the two option distributions and how each rates the other's letter (5); differences of 12 answer features	25: identical program and syntax tree; program similarity; overlap of imports, calls and raised exceptions (6); four properties in one program only; differences of 15 answer features
Logistic regression	52	56	68
Gradient-boosted tree	115	115	158

Table 5: Selector features per domain, with counts. The logistic regression sees one answer’s features and the question features; the tree sees both answers, the question and the pair. Token statistics are the mean, minimum, 10th percentile and standard deviation of the log-probability of each output token, the mean, 90th percentile and maximum entropy, the mean and 10th percentile of the margin between the two most likely tokens, and the share and longest run of tokens with probability below 0.5; the 12 statistics add the mean largest logit.

	Main	Output $\times 4$					Pairs cheaper	
	All	All	Math	Knowl.	Code	Reas.	Main	$\times 4$
<i>Relative to the stronger model alone</i>								
Stronger reviews	2.76	2.24	2.17	1.92	3.05	1.83		
Weaker reviews	2.56	2.05	2.03	1.75	2.66	1.75		
Selector	1.93	1.94	2.12	1.77	2.11	1.77		
<i>Selector relative to review</i>								
vs. stronger reviews	0.69	0.89	0.98	0.92	0.69	0.96	48	34
vs. weaker reviews	0.82	0.98	1.05	1.02	0.83	1.02	48	28

Table 6: Cost ratios under the main cost $c = N(t_{\text{in}} + t_{\text{out}})$ and with output tokens weighted four times, $c' = N(t_{\text{in}} + 4t_{\text{out}})$: the ratio is taken per pair, then averaged with domains weighted equally. Pairs cheaper: of 48, the pairs where the selector costs less than review.

	Stronger model reviews							Weaker model reviews				
	Δ_G	Δ_S	W	N	A	L	B	$\Delta_S = \Delta_G$	W	N	B	
Math	+10.3 [+9.9, +10.9]	-4.5 [-5.1, -3.9]	0	1	9	2	0	-1.3 [-1.5, -1.0]	6	3	3	
Knowledge	+9.2 [+8.5, +9.8]	-1.7 [-2.3, -1.1]	0	0	4	7	1	+0.1 [-0.3, +0.5]	4	3	5	
Code	+2.7 [+1.9, +3.6]	-3.5 [-4.6, -2.3]	1	3	3	5	0	-4.0 [-4.8, -3.2]	8	4	0	
Reasoning	+8.0 [+7.1, +9.0]	-2.7 [-3.6, -1.7]	0	1	6	5	0	-1.1 [-1.7, -0.4]	4	8	0	
All	+7.6 [+7.2, +7.9]	-3.1 [-3.5, -2.7]	1	5	22	19	1	-1.6 [-1.8, -1.3]	22	18	8	

Table 7: Section 4 by domain: mean Δ_G and Δ_S (points) over the 12 directions of each kind, with 95% intervals that resample questions, and the number of directions at each outcome level: worse than G (W), no different from G (N), above G but below the stronger model (A), level with the stronger model (L), beyond it (B). When the weaker model reviews, G is the stronger model, so $\Delta_S = \Delta_G$ and only W, N and B occur. All: domains weighted equally.

	Relative to the stronger model alone				Selector relative to review	
	Stronger reviews	Weaker reviews	Selector	JEV	vs. stronger reviews	vs. weaker reviews
Math	-4.7 [-5.4, -4.0]	-1.4 [-1.8, -1.1]	+1.2 [+0.9, +1.5]	+3.7 [+3.3, +4.1]	+5.9 [+5.2, +6.6]	+2.6 [+2.3, +3.0]
Knowledge	-1.3 [-2.1, -0.6]	+0.2 [-0.3, +0.6]	+2.1 [+1.7, +2.5]	+5.5 [+5.0, +6.0]	+3.4 [+2.7, +4.2]	+1.9 [+1.5, +2.5]
Code	-3.5 [-4.7, -2.3]	-4.0 [-4.8, -3.3]	+0.5 [-0.1, +1.0]	+2.0 [+1.2, +2.7]	+3.9 [+2.9, +5.1]	+4.5 [+3.6, +5.4]
Reasoning	-2.6 [-3.7, -1.7]	-1.1 [-1.7, -0.5]	+0.9 [+0.4, +1.3]	+5.1 [+4.4, +5.7]	+3.5 [+2.5, +4.5]	+1.9 [+1.2, +2.6]
All	-3.0 [-3.5, -2.6]	-1.6 [-1.9, -1.3]	+1.2 [+0.9, +1.4]	+4.1 [+3.8, +4.3]	+4.2 [+3.8, +4.6]	+2.7 [+2.4, +3.1]

Table 8: Section 5 by domain: accuracy differences (points) with 95% intervals that resample questions. Math and knowledge use the 1,500 held-out questions, so these numbers and intervals differ slightly from Table 7. All: domains weighted equally.

	48 pairs (96 directions)	Leaving out one model (range of the mean)	All completed pairs (131 directions)
<i>Both kinds of direction pooled</i>			
Weaker reviews – stronger reviews	+1.5 [+1.1, +1.9]	+1.2 to +1.9	+1.7 [+1.4, +2.1]
Δ_G	+3.0 [+2.7, +3.3]	+2.5 to +3.5	+3.2 [+3.0, +3.5]
Δ_S	-2.3 [-2.6, -2.0]	-2.7 to -1.6	-2.5 [-2.8, -2.2]
Net change where both agree	+0.2 [+0.0, +0.4]	+0.1 to +0.4	+0.2 [+0.0, +0.4]
Drop in disagreement with G	+14.6 [+14.1, +15.1]	+13.3 to +16.1	+14.7 [+14.2, +15.1]
Oracle over the stronger model	+7.5 [+7.2, +7.8]	+7.0 to +8.0	+7.3 [+7.1, +7.6]
<i>By kind of direction</i>			
Δ_G , stronger reviews	+7.6 [+7.2, +7.9]	+6.0 to +8.6	
Δ_S , stronger reviews	-3.1 [-3.5, -2.7]	-3.6 to -2.2	
$\Delta_S = \Delta_G$, weaker reviews	-1.6 [-1.8, -1.3]	-1.9 to -1.1	
Net change where both agree, stronger reviews	+0.9 [+0.7, +1.2]	+0.7 to +1.3	
Net change where both agree, weaker reviews	-0.5 [-0.7, -0.3]	-0.6 to -0.4	
Drop in disagreement with G, stronger reviews	+10.7 [+10.2, +11.2]	+9.4 to +11.7	
Drop in disagreement with G, weaker reviews	+18.5 [+18.0, +19.0]	+16.6 to +20.5	

Table 9: Robustness of the main averages (points, domains weighted equally, 95% intervals that resample questions). Leaving out one model removes every direction involving it; the column gives the lowest and highest mean over the 12 models. All completed pairs adds every other pair we ran review on; it is computed with both kinds of direction pooled only.

	Mean	Questions	Pairs and questions
Δ_S , stronger reviews	-3.1	[-3.5, -2.7]	[-4.1, -2.0]
Δ_S , weaker reviews	-1.6	[-1.8, -1.3]	[-2.4, -0.8]
Oracle over the stronger model	+7.5	[+7.2, +7.8]	[+6.7, +8.3]
Oracle after review, stronger reviews	+0.6	[+0.2, +1.0]	[-0.5, +1.6]
Oracle after review, weaker reviews	+2.8	[+2.6, +2.9]	[+2.3, +3.3]
Net change where both agree, stronger reviews	+0.9	[+0.7, +1.2]	[+0.5, +1.4]
Net change where both agree, weaker reviews	-0.5	[-0.7, -0.3]	[-0.9, -0.1]
Drop in disagreement with G	+14.6	[+14.1, +15.1]	[+13.2, +16.0]
Selector – stronger alone	+1.2	[+0.9, +1.4]	[+0.8, +1.6]
JEV – stronger alone	+4.1	[+3.8, +4.3]	[+3.4, +4.8]
Selector – stronger reviews	+4.2	[+3.8, +4.6]	[+3.3, +5.1]
Selector – weaker reviews	+2.7	[+2.4, +3.1]	[+2.0, +3.5]

Table 10: Main averages (points, domains weighted equally) with 95% intervals that resample questions only, as in the main text, or model pairs and questions.

G	V	Final	What happens	Stronger model reviews					Weaker model reviews				
				Math	Know.	Code	Reas.	All	Math	Know.	Code	Reas.	All
✓	✓	✓	Both right alone; review keeps it.	49.5	39.9	24.6	47.5	40.4	49.4	40.5	23.8	48.1	40.4
✓	✓	×	Both right alone; review breaks it. Harm only review can do: a choice between the two answers cannot lose these.	0.9	2.0	2.3	2.6	1.9	1.0	1.4	3.1	2.1	1.9
✓	×	✓	Only G right; review keeps G’s answer.	4.7	6.1	6.3	6.0	5.8	18.6	17.0	10.4	16.8	15.7
✓	×	×	Only G right; review loses it (the G-only part of Δ_G).	0.7	2.6	1.4	2.5	1.8	1.6	2.5	3.4	2.4	2.5
×	✓	✓	Only V right; review takes up V’s answer (the V-only part of Δ_G). Capability transfer: V alone answers these right too.	9.5	10.2	4.0	9.9	8.4	0.5	2.3	1.1	1.7	1.4
×	✓	×	Only V right; review loses V’s right answer, most often to G’s wrong one. A choice between the two answers can keep these.	10.7	9.3	9.9	9.2	9.8	4.8	6.4	6.5	6.8	6.1
×	×	✓	Both wrong alone; review gets it right. Value only review can create: neither model alone nor a choice between the two answers gets these.	2.4	3.5	2.4	3.2	2.9	0.8	1.7	1.4	1.7	1.4
×	×	×	Both wrong alone and after review.	21.7	26.4	49.2	19.0	29.1	23.2	28.2	50.2	20.5	30.6

Table 11: All eight cells: the share of questions (%) by whether G alone, V alone and the final answer are right, averaged over the 12 directions of each kind in a domain; All weights the domains equally. The three parts of Δ_G in Section 3.1 are the V-only row, minus the G-only row, plus the both-wrong-then-right row minus the both-right-then-wrong row. Columns may not sum to 100 because of rounding.

Feature	Stronger model reviews (15,364)		Weaker model reviews (5,883)	
	Odds ratio	Kept (%), Q1 to Q4	Odds ratio	Kept (%), Q1 to Q4
V’s solo output length	0.75 [0.71, 0.79]	54 / 51 / 47 / 39	0.76 [0.69, 0.83]	23 / 21 / 19 / 15
V’s mean maximum logit	1.16 [1.10, 1.22]	43 / 46 / 51 / 52	1.24 [1.11, 1.38]	16 / 18 / 21 / 23
V’s mean log-probability	1.06 [1.00, 1.13]	47 / 48 / 48 / 48	1.04 [0.93, 1.18]	18 / 17 / 20 / 21
G’s output length	1.00 [0.93, 1.07]	52 / 49 / 46 / 45	0.94 [0.84, 1.06]	21 / 20 / 19 / 18
G’s mean maximum logit	0.96 [0.91, 1.01]	48 / 47 / 49 / 48	0.96 [0.87, 1.06]	20 / 17 / 19 / 21
G’s mean log-probability	0.96 [0.90, 1.02]	49 / 49 / 47 / 47	1.02 [0.91, 1.14]	19 / 18 / 19 / 21
G’s output cut off at the limit	1.06 [0.99, 1.13]	no 47 / yes 55	1.17 [1.06, 1.30]	no 19 / yes 28

Table 12: When the reviewer keeps its own right answer, on questions where G was wrong and V right alone (number of questions in parentheses). Odds ratio: change in the odds that the final answer stays right per standard deviation of the feature, from one logistic regression per kind of direction with all seven features and direction fixed effects; features are standardised within each direction, since raw logits differ in scale across models; 95% intervals clustered by question; bold intervals exclude 1. Kept: share of these questions whose final answer stays right, by quartile of the feature within each direction (Q1 lowest). Correlations, not causal effects.