WHEN CAN HISTORICAL SCORES BE REUSED AFTER JUDGE REPLACEMENT?

Anonymous authors
Paper under double-blind review

ABSTRACT

When an LLM judge is replaced, a pipeline that stores its scores must decide whether to keep the historical scores or pay to score every object again. We ask when reuse supports a stated downstream decision policy, and when establishing that is worth its cost. In three revision-pinned open-weight replacements, better discrimination and compatible decisions come apart: a Gemma replacement that separates clean from flawed proposals more sharply newly rejects 43 of 217 pool proposals, each of which its predecessor had accepted. For fixed objectwise policies, inverting the hypergeometric test bounds disagreement on the unaudited part of a finite pool, and the bound yields an audit-budget frontier that can be checked before any new judgment is bought. With 95 objects, no audit smaller than 33 can approve a 10% tolerance, and none smaller than 74 can approve 5%. Exact top-k membership, in contrast, is not identified by a partial audit. We replay the audit on saved judgments from 14 adjacent and 4 cumulative replacements served through commercial API gateways, where a model name is a routing label rather than a verified snapshot. At 10% tolerance with a 48-object audit, the best protocol saves only 2.62% of new-judge calls on average, and restarting a full batch after a failed audit costs more than rescoring outright.

1 INTRODUCTION

Many evaluation pipelines store judge scores and act on them later. The scores decide which model outputs enter training, which proposals pass review, or which candidates receive further evaluation. When the judge is replaced, the pipeline can score every object again with the new judge or keep what it has. Rescoring repeats work already paid for, and keeping the historical scores is sound only if they still support the decisions the pipeline makes.

A better judge need not preserve those decisions. Our Gemma replacement separates proposals that reviewers rated sound from those they rated unsound more clearly than its predecessor, and it also rejects proposals the predecessor accepted. A useful correction can therefore make historical acceptances obsolete. Agreement with the new judge and agreement with an external quality label are different targets, and a migration has to state which of them it preserves.

We study *decision compatibility*: agreement between a candidate policy applied to historical scores and a reference policy applied to new-judge scores on a stated population. The policy determines which information matters. A fixed threshold needs each object's position relative to a boundary, whereas a top-k rule compares every object with the rest of the collection. Mean shifts and rank correlations describe a replacement, but the loss that matters is disagreement between the resulting decisions.

Our question is when historical scores from an old judge can be reused after replacement under a specified downstream policy. Answering it in practice means knowing which policies can be certified from a finite audit and when the certificate costs less than rescoring. Figure 1 organizes the answer around *compatibility*, the loss that reuse actually incurs; *certifiability*, what an audit can establish about that loss; and *economic usefulness*, the savings after paying for audits that pass and audits that fail.

Judge replacement has been studied as a problem of measurement validity (Yang et al., 2026). We treat it as a decision about an existing score inventory. Section 2 shows what information threshold

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

and ranking policies need and proves that exact top-k membership cannot be settled by a partial audit. Section 4 casts objectwise reuse as certification on a finite pool. The confidence bound is the classical inversion of a hypergeometric test, the calculation behind zero-acceptance sampling plans in quality control. What we add is the policy-level target, a budget frontier in Section 5 that tells a pipeline before sampling whether its audit can ever approve, and an accounting in Section 6 of what audits cost once failures and fallback procedures are included.

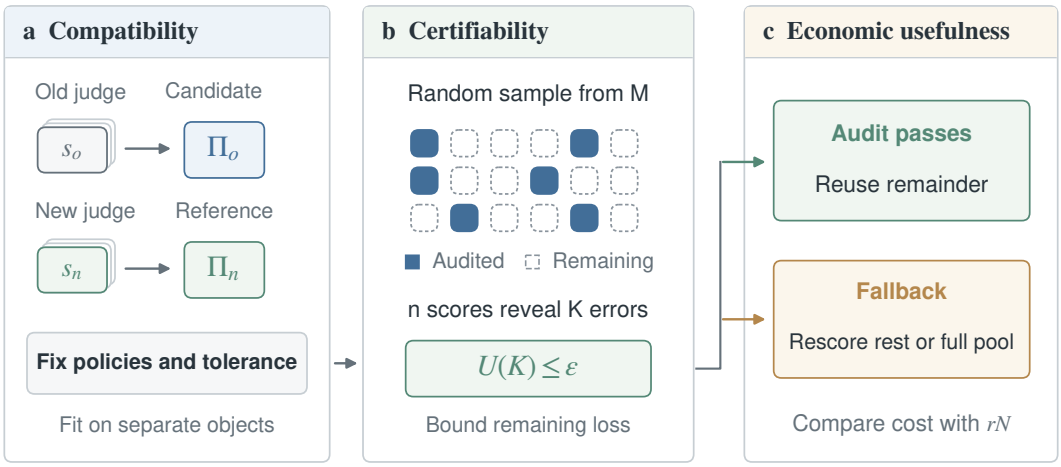


Figure 1: Historical-score reuse has three requirements. The candidate must preserve a specified decision policy, an audit must provide sufficient evidence, and the resulting deployment must justify its cost. In this objectwise workflow, fitted and audited objects use new-judge scores, and approval permits reuse only on the remaining objects. The sampling diagram is schematic.

2 What information does a reuse policy require?

Let $\mathcal{P}$ contain N objects with historical scores s_o from the old judge and scores s_n from the new judge. The candidate policy Π_o uses the historical scores, possibly after a correction fitted on other objects. The reference policy Π_n specifies the decisions the pipeline would take after full rescoring. For objectwise policies, define

$$R_{\mathcal{P}}(\Pi_o, \Pi_n) = \frac{1}{N} \sum_{i \in \mathcal{P}} \mathbf{1}\{\Pi_o(s_{o,i}) \neq \Pi_n(s_{n,i})\}. \tag{1}$$

Reuse is compatible at tolerance ε when $R_{\mathcal{P}} \leq \varepsilon$. Specifying the new judge's rubric, score aggregation and decision rule makes this target reproducible. External correctness is a different property and needs its own labels or outcomes.

For a threshold, $\Pi_\theta(s) = \mathbf{1}\{s \geq \theta\}$, one new-judge score determines an object's decision. A top-k rule uses the entire score vector. Its loss might be selection turnover, $1 - |T_o \cap T_n|/k$, or any membership change, $\mathbf{1}\{T_o \neq T_n\}$. One displaced object has small turnover but defeats exact membership preservation. The pipeline must choose its loss before assigning an audit budget.

What recalibration can preserve. A strictly increasing map h preserves all strict orders and ties in the old scores, hence the top-k set under a fixed tie rule. Threshold decisions can change when $h(s_{o,i})$ crosses the boundary. A common upward shift, for instance, can admit additional objects while preserving every ordering. A successful threshold correction therefore says nothing about ranked selection.

Nondecreasing maps can also merge distinct scores into ties. If $s_{o,i} < s_{o,j}$ but $s_{n,i} > s_{n,j}$, such a map still gives $h(s_{o,i}) \leq h(s_{o,j})$ and cannot recover the new judge's strict relation. A constant map ties every pair instead of reversing it, and no score-only correction can split objects that are already tied. Appendix A gives a conditional latent-score model for these restrictions.

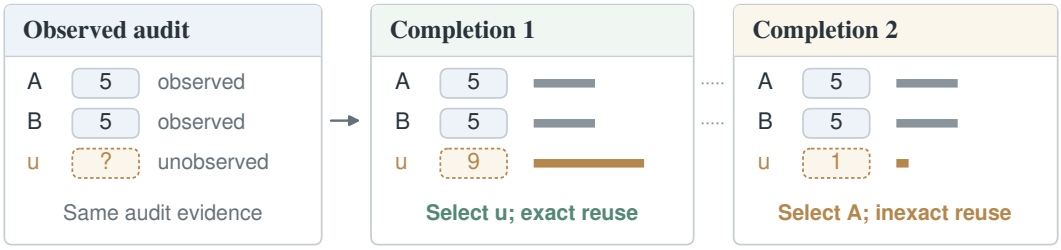


Figure 2: The missing information for exact top-k reuse. Both completions of the new judge's scores agree on audited objects A and B. With historical scores equal to completion 1 and $k = 1$, reuse is exact only in that completion. Object u is unaudited, and a fixed tie rule favors A.

What a partial audit can observe. Once a sampled object's new-judge score is available, its threshold disagreement is known. Its membership in the new top-k set may still be unresolved because an unobserved object could outrank it. Figure 2 illustrates the resulting ambiguity.

Proposition 1. Exact top-k is not identified by a partial audit. *Fix $1 \leq k < N$, a deterministic tie rule, and any proper subset $S \subsetneq \mathcal{P}$. There exist historical scores and two new-judge score vectors agreeing on S such that historical top-k reuse is exact for one vector and inexact for the other. Thus audit observations alone do not generally determine exact top-k compatibility without additional restrictions on the unobserved scores.*

Proof. Choose $u \notin S$. Give u score 9 and every other object score 5 in the first completion. In the second, change only u to score 1. Both completions agree on the audit, while u belongs to only the first top-k set. Historical scores equal to the first completion establish the claim. □

Score bounds, sufficient rescoring near the cutoff, or distributional assumptions can resolve this ambiguity, and randomized procedures close to a census may support other guarantees. The objectwise audit below requires that each sampled loss be observable, so applying it to exact top-k membership first requires resolving the missing scores in Proposition 1.

3 Diagnostics and their policy consequences

Three replacements on frozen research proposals show why the policy matters: Qwen 3.5-27B→3.6-27B, Gemma 3-27B→4-31B, and Mistral-Small 3.1-24B→3.2-24B, each with open weights pinned by revision. The collection contains 300 research proposals that Ho et al. (2026) reconstructed from ICLR submissions and paired with reviewer soundness scores. Forty-eight of them are *anchors*, a fixed reference set of 12 flawed proposals with low reviewer soundness, each matched by subfield to a clean one with high soundness, and 24 further proposals; the others form pools on which decisions are compared. Appendix B records the prompts, sample accounting and score distributions.

The diagnostics describe several properties of the reported scores. Location captures the average shift, ordering captures relations between objects, and resolution records how many distinctions survive the score interface. Clean/flawed separation measures discrimination on the anchors. These properties change in different combinations: Qwen shifts toward higher scores, Gemma improves discrimination while reordering proposals, and Mistral concentrates its outputs at one value. Figure 3 sets threshold decisions beside the ordering information that calibration preserves.

Gemma is a clear case of improved discrimination with incompatible decisions. The replacement newly rejects 43 of 217 pool objects, more than either same-judge repeat changes, and anchor calibration only reduces the count to 36. The newly rejected proposals have lower mean reviewer soundness than those accepted by both judges. This external evidence is consistent with a useful correction, although reviewer scores are themselves imperfect. Retaining the historical acceptances would give up part of the change that motivates adopting the new judge.

Mistral shows how a repair can succeed for one policy and fail for another. Its new judge assigns one value to 93% of the full non-anchor pool. Calibration matches every threshold decision in the held-out pool by placing all scores above the acceptance boundary, yet it supplies none of the distinctions

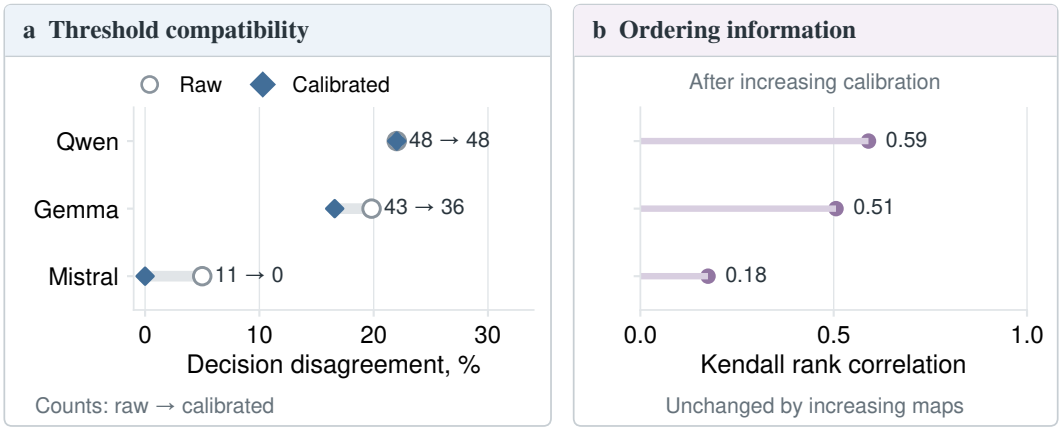


Figure 3: Calibration has policy-specific effects. Left: disagreements at threshold 6 before and after increasing affine calibration on the 48 anchors, over 218 Qwen, 217 Gemma and 220 Mistral pool objects. Right: the same correction leaves Kendall τ unchanged, so Mistral's matched threshold decisions come with weak ordering. Qwen mixes two runtimes. Table 1 adds same-judge repeats and tie-break ranges.

that top-ranked selection needs. Its wide turnover interval reflects the many selections that tied scores allow. Isotonic and affine fits produce identical threshold decisions in all three comparisons, so the contrast between policies does not depend on the choice of fitted correction.

Attribution is weaker for Qwen. Pool calls missing from its saved vLLM logs (Kwon et al., 2023) were made in the Transformers library (Wolf et al., 2020) at the same pinned revisions and settings. Of 219 available scores from each judge, 110 old-judge scores and 98 new-judge scores came from vLLM, and the paired intersection contains 218 objects. The Gemma and Mistral comparisons therefore carry the central policy contrasts. Appendix B gives the runtime accounting, and Table 2 gives the estimates saved at collection time.

Diagnostics locate possible migration problems, but reuse needs evidence about the intended decision loss. Appendix C reports our other diagnostics: shared evidence packets, alternative score interfaces, gateway comparisons on the same proposals, and a second domain.

4 Certifying objectwise reuse on a finite pool

The audit keeps fitting a candidate apart from deciding whether to trust it. Let f objects be reserved for any policy fitting, leaving $M = N - f$ objects in the audit population. Fix the candidate Π_o and reference Π_n before sampling from this population. Conditional on the fitting data and the realized new-judge scores, object i has a fixed disagreement label $Z_i = \mathbf{1}\{\Pi_o(s_{o,i}) \neq \Pi_n(s_{n,i})\}$. A simple random sample of $n < M$ objects reveals K disagreements. If the population contains D in total, then $K \sim \mathrm{Hypergeom}(M, D, n)$.

The sample need only be uniform given the potential new-judge scores, so scores can be bought as sampled.

We use the classical duality between tests and confidence bounds (Lehmann & Romano, 2005). Write $F_D(k) = \Pr_D(K \leq k)$ and define

$$D_U(k) = \max\{D \in \{k, \ldots, M - n + k\} : F_D(k) \geq \alpha\}, \qquad U(k) = \frac{D_U(k) - k}{M - n}. \tag{2}$$

Lower-tail inversion excludes error counts for which the observed audit would be unusually favorable. Subtracting the k observed errors from the retained maximum D_U bounds errors among the unaudited objects eligible for reuse.

Proposition 2. Finite-pool objectwise certificate. *Conditional on fixed scores and disjoint fitting data, the remaining disagreement* $R_{\mathrm{rem}} = (D - K)/(M - n)$ *satisfies* $\Pr\{R_{\mathrm{rem}} > U(K)\} \leq \alpha$.

Approving only when $U(K) \le \varepsilon$ therefore gives

$$\Pr\{\text{approve } \textit{and } R_{\text{rem}} > \varepsilon\} \le \alpha. \tag{3}$$

For intuition, the true D is excluded only when its own lower-tail probability is below α, and under random sampling that event has probability at most α. An erroneous approval is contained in this event, because a remaining loss above ε must also exceed an approved bound. The labels are fixed and the sample is randomized, so no independence between objects' disagreement labels is required. Appendix D gives the full proof and implementation checks.

What approval authorizes. Audited and fitting objects already have new-judge scores and use those decisions. Approval authorizes the candidate policy only on the other $M - n$ objects, and a failed audit sends them to the chosen fallback. The tolerance thus describes the portion of this hybrid deployment that was not rescored. Averaging its errors over the whole collection would give a smaller number, because it would also count objects that were already rescored.

The guarantee concerns erroneous approval over repeated audit draws. It allows the conditional fraction $\Pr\{R_{\text{rem}} > \varepsilon \mid \text{approve}\}$ to exceed α when approvals are rare.

A worked audit. Take $M = 95$, $n = 48$ and $\alpha = .05$. Zero observed disagreements permit four errors among the 47 unaudited objects, which passes a 10% tolerance but fails 5%. One disagreement raises the bound to $6/47$ and prevents either approval. Figure 4a contrasts sample rates with these bounds. A failed audit means the evidence is insufficient, even when the actual remaining loss is small.

5 Planning an audit before purchasing judgments

Some budgets cannot approve reuse under any audit outcome, and this can be checked before collecting scores. Let $m = M - n$ and $b = \lfloor \varepsilon m \rfloor$, the largest acceptable number of remaining disagreements. After observing k errors, the first unacceptable total is $D = k + b + 1$. Since $F_D(k)$ is nonincreasing in D, Equation 2 gives

$$U(k) \le \varepsilon \quad \Longleftrightarrow \quad F_{k+b+1}(k) < \alpha. \tag{4}$$

The inequality is strict because population counts whose lower-tail probability equals α are retained. Define the largest permissible audit count as

$$K_{\max}(n; M, \varepsilon, \alpha) = \max\{k \in \{0, \dots, n\} : F_{k+b+1}(k) < \alpha\}, \tag{5}$$

using $K_{\max} = -1$ when no count qualifies. Approval occurs exactly when $K \le K_{\max}$. Appendix E proves that the allowable counts form this prefix, which makes zero disagreements the most favorable observation.

At $K = 0$, the boundary population contains $b + 1$ errors, and all of them must fall outside the sample for the audit to observe none. That probability gives the first feasible budget,

$$n_{\min} = \min_{1 \le n < M} \left\{ n : \frac{\binom{M-n}{\lfloor \varepsilon(M-n) \rfloor + 1}}{\binom{M}{\lfloor \varepsilon(M-n) \rfloor + 1}} < \alpha \right\}, \tag{6}$$

provided the set is nonempty. The frontier is a design consequence of the classical bound. It distinguishes a budget that could approve from one that cannot, and actual approval still depends on the disagreements observed.

For $M = 95$, the first zero-error audit that can approve a 10% tolerance has 33 objects. Tightening the tolerance to 5% raises that number to 74, so an audit of 48 can never pass the stricter requirement, however closely the judges agree. At $M = 1000$ the corresponding sizes are 29 and 60. Figure 4c shows this dependence on pool size: as the pool grows, a feasible audit leaves a much larger share of the historical inventory available for reuse.

The frontier has discrete jumps because the target concerns the unaudited pool. Increasing n shrinks that pool and sometimes lowers its integer error allowance b. For $M = 20$, $\varepsilon = .30$ and $\alpha = .05$, a zero-error audit passes at $n = 16$ but fails at $n = 17$. Thus $n_{\min}$ marks the first feasible size rather

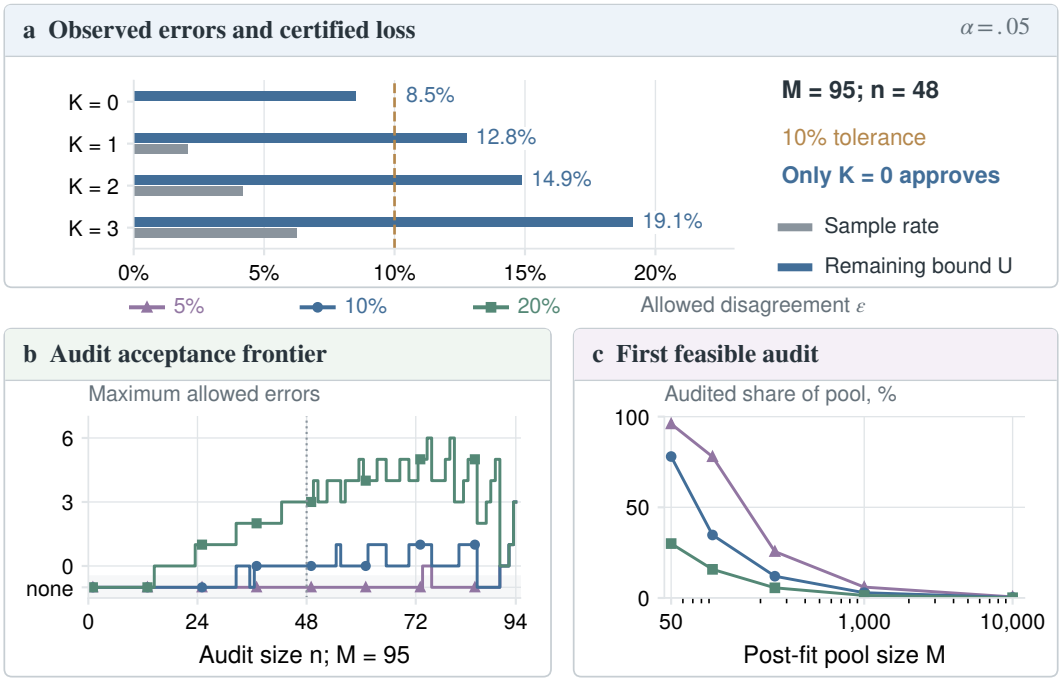


Figure 4: Certifiability at $\alpha = .05$. Panel a contrasts observed disagreement with the upper bound on the 47 unaudited objects. Panel b shows which counts permit approval at $M = 95$; "none" excludes every outcome. Panel c gives the first feasible audit fraction. Tables 7 and 8 list the numerical values.

than a threshold above which every budget qualifies, and a planned size should be checked directly through $K_{\max}$.

This check supports an early operational decision. If the planned audit has $K_{\max} = -1$, the pipeline can avoid buying evidence that cannot satisfy its rule, and it must change the budget, reconsider whether the tolerance is justified, or rescore the collection. When $K_{\max} \geq 0$, the question becomes economic: how often the audit will pass, and how much work will then be left to save.

6 When does certification save work?

6.1 Cost depends on the fallback procedure

Treat historical calls as sunk and let each new-judge score require r replicate calls. Full rescoring costs $C_{\text{full}} = rN$. Fitting on f objects and auditing n more costs $C_0 = rf + rn$ before a reuse decision is reached. If these stages use different scoring protocols, the terms become $r_f f + r_a n$, and any missing reference repeats must also be paid for. Our empirical calculation uses the median of three judgments at every stage, so $r_f = r_a = r = 3$.

Let A indicate approval and $m = N - f - n$ be the number of objects still unjudged. Under *continuation fallback*, a failed audit triggers rescoring of those m objects while keeping the judgments already obtained. Under *full-batch fallback*, failure launches an additional pass over all N objects, as in a pipeline that requires a fresh complete batch:

$$C_C = r(f + n) + (1 - A)rm = C_{\text{full}} - Arm, \tag{7}$$

$$C_B = r(f + n) + (1 - A)rN. \tag{8}$$

For a fixed protocol with approval probability p, expected savings are prm and $prN - r(f + n)$, respectively. Continuation can save at most rm calls, whereas full-batch fallback saves work only if $p > (f + n)/N$. A fixed orchestration overhead h subtracts h from either saving. We count calls because their accounting is explicit; token charges, latency and monetary cost would need further measurements.

For a fixed candidate with D disagreements among the M audit-pool objects, the exact approval probability and continuation savings are

$$p_D = \Pr_D\{K \le K_{\max}\}, \qquad \mathbb{E}_D[C_{\text{full}} - C_C] = r(M-n)p_D, \tag{9}$$

with $p_D = 0$ for an infeasible budget. These expressions support design calculations under stated drift scenarios. With $M = N = 1000$, no fitting and $D = 0$, for instance, an audit of 29 objects meets a 10% tolerance at $\alpha = .05$ and saves 97.1% of calls. This best case assumes no drift.

Approval coverage and savings reward different budget choices. Larger audits approve more often but leave fewer objects unjudged, and fitting can reduce disagreement while consuming judgments and shrinking the reusable population. The first feasible size $n_{\min}$ therefore need not minimize expected cost, and Equation 9 lets a pipeline compare budgets under several assumed drift levels before committing to one.

6.2 Retrospective design and call accounting

The cost analysis replays 18 saved gateway comparisons, 14 between adjacent releases and 4 from the first to the last release of a series, with 300 seeded permutations each and no new judge calls. A score-blind hash selects one implementation per task, and complete three-repeat records leave 111 to 119 tasks per comparison. Tasks, configurations and permutations are shared across protocols, so the $5{,}400$ resplits per condition are dependent and describe the saved records rather than independent deployments.

The reference policy accepts a new-judge median score of at least 7. We compare three candidates: the unchanged threshold with the same 24 objects reserved, a threshold fitted on those objects, and the unchanged threshold with no fitting reservation. Fitting minimizes disagreement over integer cutoffs 1 to 11, with ties broken toward 7 and then toward the closest cutoff. The reserved unchanged policy allows a comparison on identical audit populations, and the version without fitting measures what the policy costs when no judgments are spent on fitting.

Each audit uses a uniform subset disjoint from fitting. Purchased scores determine the decisions for fitting and audited objects, the candidate handles approved remaining objects, and fallback supplies new-judge decisions after failure. We count three calls per score throughout. Full-batch fallback is a cost scenario evaluated against the same saved target and does not simulate fresh stochastic calls. Canonical unit tests provide an external correctness label.

At $n = 48$ and a 10% tolerance, unchanged reuse with 24 reserved objects passes in 209 of $5{,}400$ resplits, and 19 of the approved pools exceed tolerance. The conditional fraction is $19/209 = 9.1\%$, while erroneous approvals account for $19/5{,}400 = 0.35\%$ of all resplits. Proposition 2 controls the latter quantity.

The plug-in rule $K/n \le \varepsilon$ approves far more often, $2{,}941$ times in the same resplits, but 898 of those approvals exceed tolerance. Approval coverage alone therefore cannot rank reuse procedures, which must first be compared under the same error requirement, and our cost calculations use the finite-pool certificate throughout. Appendix F gives the full comparison.

6.3 Savings, budget, and external correctness

Under continuation fallback, unchanged reuse with 24 reserved objects saves 5.5 calls on average from a full-rescore budget of 353.7, or 1.54%. Threshold fitting approves less often and saves 1.22%. Removing the fitting reservation from the unchanged policy raises savings to 2.62%. At this budget, none of the three protocols ever approves at 5% tolerance. Allowing 20% disagreement gives a different operating point, where unchanged reuse without fitting saves 27.6% of calls; Table 9 shows the loss that comes with it.

The fallback can change the sign of the saving without changing a single approval. At 10% tolerance, the protocol without fitting incurs 128.5 additional calls on average if failure triggers a full batch, whereas at 20% it saves 20.1 calls under the same fallback. The statistical certificate is identical in both execution models, so the economic outcome is decided by whether the pipeline keeps the judgments it bought during a failed audit.

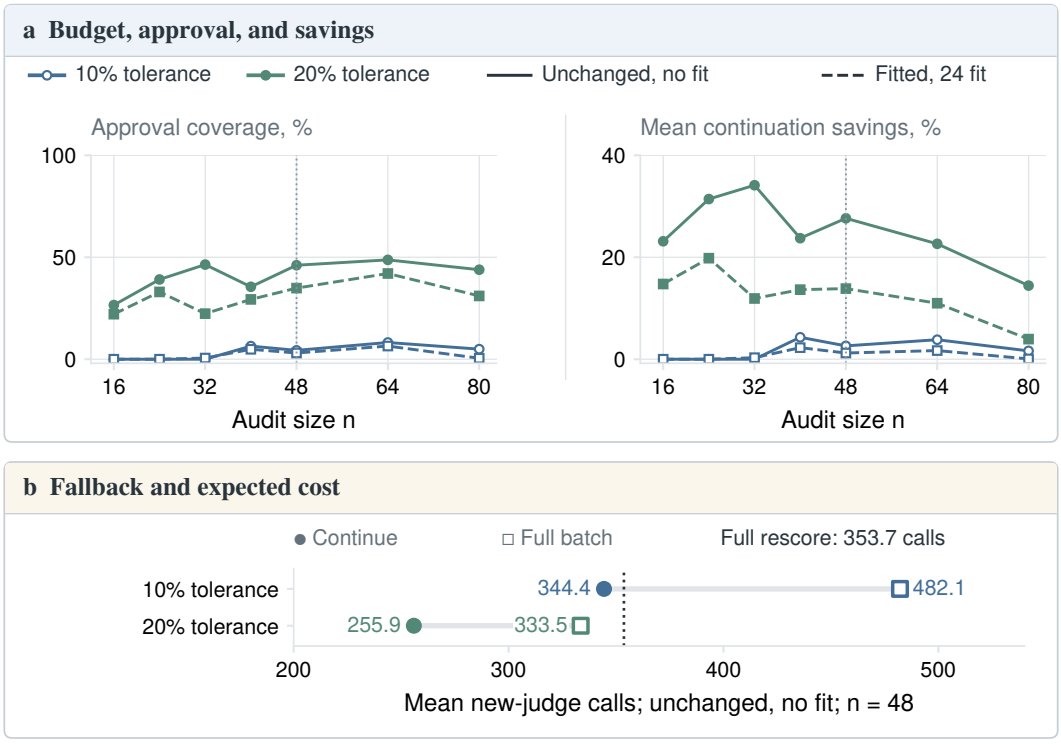


Figure 5: Economic usefulness depends on the audit budget and fallback. Panel a compares approval coverage with continuation savings. Panel b holds the audit without fitting fixed at $n = 48$: a full batch on failure can cost more than rescoring without an audit. Each point summarizes $5{,}400$ dependent retrospective resplits for one fixed protocol. Table 9 reports all three candidates at this budget, including remaining disagreement.

Mean continuation savings at 10% tolerance stay below 5% for every evaluated budget and candidate, with a maximum of 4.3%. Figure 5 shows why coverage and savings diverge: each approved audit leaves a different number of reusable scores.

The average also hides differences between migrations. At $n = 48$ and 10% tolerance, 10 of the 18 comparisons never approve without fitting in the saved resplits, while one saves 22.9% of calls under continuation and another saves 11.3%. Both are adjacent Gemini comparisons on the same code population. Appendix F reports every configuration.

Approval does not make compatibility and external correctness the same thing. For reuse without fitting at $n = 48$ and 10% tolerance, approved remaining pools have mean disagreement of 4.6% and unit-test decision accuracy 0.25 percentage points below the new policy. The complete pipeline has a smaller gap because judged objects and failed audits use new-judge decisions. Appendix F reports both denominators.

7 Related Work

Judge change and drift. Yang et al. (2026) audit judge reliability across upgrades, including accuracy, biases and multi-judge mechanisms. Singh et al. (2025) study the compatibility of fine-tuned judges across generator generations, and Li (2026) attributes drift to the evaluated system or to the judge. These works ask whether a judge's measurements change, and Zhang et al. (2025) find that human experts who rate the same research ideas twice also shift their standards. We ask which stored scores may stay in use under a stated loss, and what evidence justifies keeping them.

Measurement and calibration. Rater-effect and item-response models characterize judges as measurement instruments (Casabianca & Beiting-Parrish, 2026; Choi et al., 2026), and calibration

methods correct their scores (Morandi, 2026). Such models supply candidate reuse policies, which the audit then accepts or rejects on a stated population and decision loss.

Label-efficient evaluation and risk control. Active testing chooses which test points to label so that a model's risk can be estimated from few labels (Kossen et al., 2021), and prediction-powered inference combines a small labeled sample with model predictions to obtain valid confidence intervals (Angelopoulos et al., 2023). Both target a population quantity. The reuse decision instead needs a one-sided bound on disagreement among the specific unaudited objects whose historical scores will be kept, with a uniform sample so that the bound requires no model of either judge. Learn-then-test (Angelopoulos et al., 2025) calibrates a decision rule so that its risk on future exchangeable data is controlled with high probability, and conformal risk control (Angelopoulos et al., 2024) bounds that risk in expectation; our target is the realized loss on a fixed finite pool.

Acceptance sampling. The certificate itself is classical. Inverting the hypergeometric test (Lehmann & Romano, 2005) gives exact confidence bounds for finite populations (Wright, 1991), and plans that reject a lot on the first observed defect are standard in quality control (Guenther, 1969; Squeglia, 2008). Our contribution is to connect these tools to the information each decision policy requires, to the unaudited pool that reuse actually affects, to a budget frontier that can be checked before scoring, and to the cost of acting on a failed audit.

8 Scope and implications for an evaluation pipeline

Figure 1 gives the deployment sequence: specify the source scores, reference policy, population and tolerated disagreement; fit any correction on other objects; then check the proposed budget through $K_{\max}$ and price the fallback. Even a feasible budget may be unattractive when approval is unlikely or little work remains to save. The verdict applies to one inventory and one policy, and checking the immediately preceding judge does not establish compatibility with an older source, as Appendix G illustrates on two release series. Recording the source configuration, rubric, score aggregation, sampled objects and tolerance with the verdict makes its scope inspectable at the next migration.

Limitations. The certificate holds under its stated conditions: both policies are fixed before sampling, fitting uses other objects, and the audit sample is uniform. Choosing a policy or budget after inspecting audit errors requires a fresh sample or simultaneous error control, and adaptive or sequential audits need their own error control. The binary loss weights every disagreement equally, and a pipeline that penalizes false acceptance more heavily needs a different target and bound. Compatibility with the new judge is not external correctness, and a future population needs its own evidence. Exact top-k selection needs further score information or assumptions. On the empirical side, the proposal experiments use pinned open weights with the Qwen runtime mixture described above. Gateway labels identify deployed configurations whose hidden prompts, tokenization or routing may differ, so these comparisons support accounting at that level rather than attribution to weight updates. The code judgments come from one generator, the audits concern complete-case pools in which missing scores may be informative, and shared tasks and configurations limit replication. Appendix H records which analyses were specified before data collection and where recorded dates cannot confirm it.

9 Conclusion

Historical scores can still serve a pipeline after a judge replacement when they preserve the decisions it intends to make, and what must be preserved depends on the policy. For fixed objectwise rules, a uniform audit of the pool gives a classical bound on erroneous approval, and the same bound tells a pipeline in advance whether a planned audit can ever approve. Ranking policies need information that a partial audit does not supply. On the saved gateway judgments, certification behaved as intended but saved little, because the pools were small and many migrations drifted beyond a strict tolerance, and a full-batch fallback turned the audit into a net cost. A pipeline considering reuse should check the frontier and price its fallback before it buys the first new judgment.

Reproducibility statement

The paper and its appendices contain what is needed to check each result: the models with their revisions or gateway labels, decoding settings, prompts and parse rules (Appendix B.1), the populations and sample accounting (Appendices B and H), the audit and its proofs (Appendices D and E), and the per-comparison cost results (Appendix F). Scripts and records that reproduce these numbers are included in the supplementary material; no step makes a model call. From the records, the scripts regenerate the audit, frontier and cost results of Sections 4 to 6 and Appendices D to F, the gateway-study tables of Appendix G, the decision comparisons of Table 1, and Figures 3 to 5 and 8 to 10; the same scripts draw the schematic Figures 1 and 2. The other proposal-study values, including Figures 6 and 7, Tables 2, 4 and 6 and the bootstrap intervals, are read from saved per-phase summaries, which are included with the call logs. The records contain requested model labels, raw outputs and parsed scores for the two gateway studies: $14{,}952$ unique pointwise calls, stored in $15{,}786$ lines because a resumed shard wrote some calls twice (Appendix H), and $2{,}408$ pairwise calls in the first study, and $8{,}443$ parsed calls in the second. The proposal study made 14,882 calls, 13,646 of them to open-weight models. Some early call logs of the Qwen pair are incomplete in the saved records, so the values that depend on them are read from the summaries. API labels do not identify provider weights. The certifiability and cost analysis checks the hashes of its input records and verifies the finite-population bounds by exact enumeration.

Ethics statement

The study scores pinned open-weight model revisions and gateway-labeled deployments on public research proposals and on programs written for public programming problems, and involves no human subjects. The title-masking manipulation could be read as a way to evade contamination checks; we use it only as a diagnostic with matched controls, described in Appendix C.1.

LLM usage

LLMs are the objects of study here, and as part of the method they also generated some study materials, namely fact packets, defect summaries, masked titles, text revisions and rationale codes, which Appendix C describes. Beyond that role, the authors used LLM assistants to edit and help draft the text. The authors designed the experiments, checked the reported numbers against the saved records and take responsibility for the content.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, et al. Phi-3 technical report: A highly capable language model locally on your phone. arXiv:2404.14219, 2024a.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, et al. Phi-4 technical report. arXiv:2412.08905, 2024b.

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

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations*, 2024.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

Jodi M. Casabianca and Maggie Beiting-Parrish. Correcting human labels for rater effects in AI evaluation: An item response theory approach. arXiv:2602.22585, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, et al. Evaluating large language models trained on code. arXiv:2107.03374, 2021.

Junhyuk Choi, Sohhyung Park, Chanhee Cho, Hyeonchu Park, and Bugeun Kim. Diagnosing the reliability of LLM-as-a-judge via item response theory. In *International Conference on Machine Learning*, 2026.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. arXiv:2606.19348, 2026.

Gemma Team. Gemma 3 technical report. arXiv:2503.19786, 2025.

William C. Guenther. Use of the binomial, hypergeometric and Poisson tables to obtain sampling plans. *Journal of Quality Technology*, 1(2):105–109, 1969.

Sy-Tuyen Ho, Minghui Liu, Huy Nghiem, and Furong Huang. SoundnessBench: Can your AI scientist really tell good research ideas from bad ones? arXiv:2605.30329, 2026.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, et al. Qwen2.5-Coder technical report. arXiv:2409.12186, 2024.

Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. Active testing: Sample-efficient model evaluation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5753–5763, 2021.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165.

Erich L. Lehmann and Joseph P. Romano. *Testing Statistical Hypotheses*. Springer, 3 edition, 2005. doi: 10.1007/0-387-27605-X.

Yitao Li. Who drifted: the system or the judge? anytime-valid attribution in LLM evaluation pipelines. arXiv:2606.15474, 2026.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522, 2023. URL `https://aclanthology.org/2023.emnlp-main.153/`.

Andrea Morandi. Two ways to de-bias an LLM-as-a-judge: A continuous-score comparison of hierarchical Bayesian calibration and neural-ODE score transport. arXiv:2605.09227, 2026.

Niklas Muennighoff, Qian Liu, Armel Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro von Werra, and Shayne Longpre. OctoPack: Instruction tuning code large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Janvijay Singh, Austin Xu, Yilun Zhou, Yefan Zhou, Dilek Hakkani-Tür, and Shafiq Joty. On the shelf life of fine-tuned LLM-judges: Future-proofing, backward-compatibility, and question generalization. arXiv:2509.23542, 2025.

Nicholas L. Squeglia. *Zero Acceptance Number Sampling Plans*. ASQ Quality Press, 5 edition, 2008.

Team GLM. ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools. arXiv:2406.12793, 2024.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, 2020. doi: 10.18653/v1/2020.emnlp-demos.6.

Tommy Wright. *Exact Confidence Bounds when Sampling from Small Finite Universes: An Easy Reference Based on the Hypergeometric Distribution*, volume 66 of *Lecture Notes in Statistics*. Springer, 1991.

Zongyou Yang, Yinghan Hou, and Xiaokun Yang. When the judge changes, so does the measurement: Auditing LLM-as-judge reliability. arXiv:2607.08535, 2026.

Lingyu Zhang, Mitchell Wang, and Boyuan Chen. Scientific judgment drifts over time in AI ideation. arXiv:2511.04964, 2025.

A MEASUREMENT MODEL

One illustrative parameterization shows how location, ordering and resolution can change in different combinations:

$$s_v(x) = Q_v\big[g_v\big(w_v^\top z_v(x) + b_v\big)\big]\,, \tag{10}$$

where $z_v(x)$ is a version-specific assessment of object x, w_v aggregates it, b_v sets location, g_v is a response map and Q_v is the reported score interface, including its discretization. A change in b_v can shift location, a change in w_v or z_v can reorder objects, and a coarse Q_v or a saturated g_v can reduce resolution. We use Equation 10 to read the observed relations and do not attempt to identify or estimate its latent components.

The result behind the repair restrictions of Section 2 is the following. Each judge version v scores an object x as $s_v(x) = g_v(w_v^\top z(x) + b_v)$, where $z(x) \in \mathbb{R}^K$ is a shared reference representation used only in this conditional model, $w_v \in \mathbb{R}^K \setminus \{0\}$, $b_v \in \mathbb{R}$, and g_v is strictly increasing. Write o and n for the old and the new version and $Z = \{z(x) : x \in \mathcal{X}\}$ for the realized set of assessments. We call Z rich if the difference set $Z - Z$ meets every nonempty open cone in $\mathbb{R}^K$; it suffices, for instance, that Z contains an open ball.

Proposition 3. Global recalibration requires preserved geometry. *Assume Z is rich. (i) If $w_n = c\,w_o$ for some $c > 0$, then $s_n = h \circ s_o$ on all of $\mathcal{X}$ for the strictly increasing map $h(s) = g_n(c(g_o^{-1}(s) - b_o) + b_n)$. (ii) If w_n is not a scalar multiple of w_o, then there exist pairs whose order the two versions agree on and pairs whose order they reverse. No monotone h, increasing or decreasing, then satisfies $s_n = h \circ s_o$ on $\mathcal{X}$, and every monotone recalibration of old scores fails to recover at least one strict order relation of the new judge. (iii) If $w_n = c\,w_o$ with $c < 0$, the strictly decreasing h of the same form applies, and the versions agree only up to a global inversion of the scale.*

Proof. For (i) and (iii), $g_o^{-1}(s_o(x)) - b_o = w_o^\top z(x)$, so $h(s_o(x)) = g_n(c\,w_o^\top z(x) + b_n) = s_n(x)$, and h is strictly increasing for $c > 0$ and strictly decreasing for $c < 0$. For (ii), the cones $C_\pm = \{u : w_o^\top u > 0,\ \pm w_n^\top u > 0\}$ are open, and both are nonempty: there is a $u_\perp$ with $w_o^\top u_\perp = 0$ and $w_n^\top u_\perp > 0$, and for any u_0 with $w_o^\top u_0 > 0$ the vector $u_0 + t\,u_\perp$ lies in C_+ and $u_0 - t\,u_\perp$ in C_- for large t. Richness supplies $z_i - z_j \in C_+$ and $z_k - z_l \in C_-$. Because g_o and g_n are strictly increasing, the first pair keeps its order under both versions and the second pair reverses it. An increasing h fails on the second pair, and a decreasing h fails on the first. For a nondecreasing or nonincreasing map, the contradicting strict order is either reversed or merged into a tie. □

Corollary 1. What a global map can absorb. *Assume Z is rich and $g_o = g_n = \mathrm{id}$. The version gap $\Delta(x) = s_o(x) - s_n(x) = (w_o - w_n)^\top z(x) + (b_o - b_n)$ is constant across objects if and only if $w_o = w_n$. An affine correction absorbs the constant term and the component parallel to w_o, and noncollinear weights still produce both preserved and reversed pairs, whose strict relations under the new judge no monotone recalibration can all recover.*

The observed patterns motivate this conditional model without identifying its latent cases. Qwen's mean correction on representative pools leaves residual dispersion, Gemma changes the order and the sign of the gap across quality strata, and Mistral concentrates its reported score. The paper uses only the part of the model that can be checked directly: an increasing map cannot reverse a pair, and reproducing a constant score does not restore discrimination.

Table 1: Decision changes on the held-out pool at threshold 6. Increasing affine calibration is fitted on the 48 anchors. Top-30% turnover spans the tie-breaks allowed by the scores. Qwen's pool scores combine vLLM calls with calls completed in Transformers, and its τ and gap differ from the estimates of Table 2, which were saved at collection time.

	Qwen	Gemma	Mistral
Complete objects	218	217	220
Cross-judge Kendall τ	0.590	0.506	0.175
Standardized gap Δ_{ver}	-0.404	$+0.110$	-0.450
Threshold disagreements	48/218	43/217	11/220
newly accepted / rejected	41 / 7	0 / 43	11 / 0
After calibration	48/218	36/217	0/220
Same-judge disagreements	11/105, 17/91	1/217, 12/217	10/220, 0/219
Top-30% turnover	12 to 57%	12 to 75%	3 to 100%

B PROPOSAL STUDY: DESIGN AND DIAGNOSTIC RESULTS

Populations. The three open-weight replacements of Section 3 are Qwen 3.5-27B→3.6-27B, Gemma 3-27B→4-31B and Mistral-Small 3.1-24B→3.2-24B, each pinned by revision hash (Table 3) and analyzed separately. The proposal study scores 300 frozen objects derived from ICLR submissions from 2022 to 2026, with reviewer soundness scores and verified risk factors taken from Ho et al. (2026). A seeded script drew the anchors from the benchmark without editing their text, and no model generated or rewrote them. A flawed anchor has reviewer soundness of at most 1.5 and at least two verified risk factors, a clean anchor soundness of at least 3.5. Each of the 12 flawed anchors is matched to one clean proposal from the same subfield, or from the same broad research area in 3 pairs, preferring the same submission year, which holds in 8 pairs, and then the closest text length. The other 24 anchors are six from each of four research areas, three with low and three with high reviewer soundness, spread over submission years. The 300 objects are the 48 anchors, 32 objects chosen from 96 screening candidates by a recognition probe (20 recognized only by the new Qwen and 12 matched controls; the other 64 candidates received only the probe), and a 220-object held-out pool. The anchors and the 32 chosen objects form the 80-object set for the evidence interventions of Appendix C.1, and the 252 non-anchor objects form the full pool. The prompt at the heart of every comparison, called L0, contains the rubric and the proposal and asks for an integer score from 1 to 10; Appendix B.1 gives it verbatim. Prompt-repeat reliability at L0 is a quadratic-weighted κ between 0.86 and 0.93.

The code population used in Section 6 and Appendix G holds 356 Python functions sampled from Qwen2.5-Coder-7B-Instruct (Hui et al., 2024) and labeled by execution against canonical unit tests. Its 86 anchors are 43 matched pairs, each with a passing and a failing solution to one problem, and its 270 held-out implementations include 107 from tasks also represented among the anchors. The functions are completions of the 164 HumanEval problems (Chen et al., 2021) by Qwen2.5-Coder-7B-Instruct at temperature 1.0 and top-p 0.95 with seed 20260920, six per problem; of 984 generations, 977 yielded an extractable function, 799 of which pass the canonical tests. Judges do not execute the code. The judges were reached through commercial API gateways, and model names are the gateways' routing labels, not independently verified provider snapshots. We shorten them, writing Opus 4.6 for `claude-opus-4-6` or Gemini 3.5 Flash for `gemini-3.5-flash`, and Appendix B.1 lists the exact labels. Reported prompt-token counts differ across labels, 122 for Opus 4.6 against 169 for later labels, so we cannot verify that the effective inputs were identical.

Standardized gap. For a prompt condition C, the standardized gap is $\Delta_{\text{ver}}(C) = \overline{(s_{\text{old}} - s_{\text{new}})}/\sigma_{L0}$, where σ_{L0} is the pooled L0 standard deviation of that replacement, so a positive value means the new judge is stricter. Intervals resample objects with 20,000 bootstrap draws.

Location, ordering and discrimination. Figure 7 and Table 2 summarize the three replacements. Qwen shifts toward higher scores, by 0.31 SD on the anchors and 0.33 SD on the pool, with both intervals excluding zero, and separates clean from flawed anchors more sharply than its predecessor, 3.17 SD against 1.88. Its cross-version ordering nonetheless falls below the repeat stability of either version, so a location correction alone does not address its rank changes. Gemma's shift toward

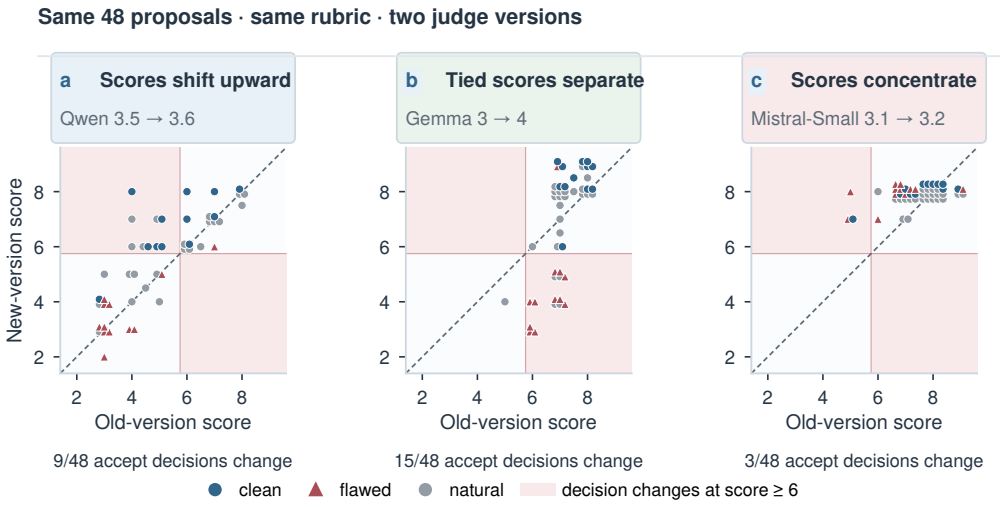


Figure 6: The 48 anchor proposals scored by the old and the new version of three judges. Each mark is one proposal at its median score over repeated calls, with marks that share a score spread slightly within the cell. Clean anchors are blue circles, flawed anchors red triangles and the other 24 anchors grey. Proposals in the shaded quadrants change their acceptance decision at a threshold of 6.

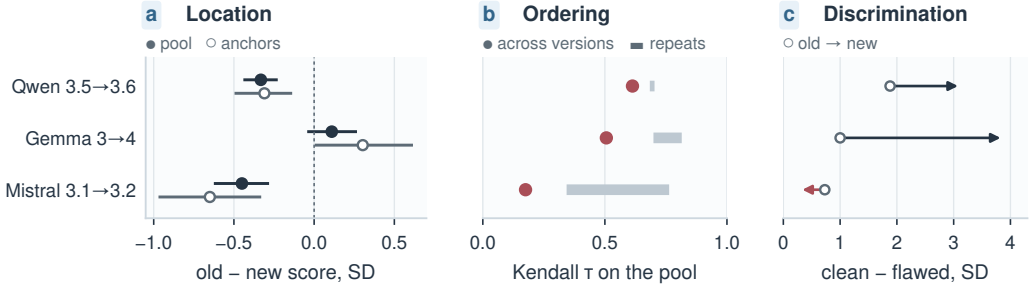


Figure 7: Location, ordering and discrimination for the three replacements. Panel a: standardized old-minus-new gap on the pool and on the anchors, with 95% bootstrap intervals; negative values mean the new version is more lenient. Panel b: cross-version Kendall τ on the pool against the range of the two versions' same-version repeat τ. Panel c: separation of clean from flawed anchors under each version.

lower scores of 0.30 SD on the anchors shrinks to 0.11 SD on the pool, where the interval includes zero, and the average hides a reversal across reviewer-rated quality: from the weakest to the strongest tertile of reviewer scores the gap is $+0.80$, -0.20 and -0.43 SD. Its separation rises from 1.00 to 3.92 SD. Mistral loses resolution: one value accounts for 93% of full-pool scores, 86% of object pairs are tied, and separation falls from 0.73 to 0.24 SD. The sign of each pool gap is the same in all five submission years from 2022 to 2026, which does not establish that the objects are absent from any model's training data.

Same-version baselines. The Qwen and Gemma judges scored each of the 220 held-out pool objects a second time, and the two Mistral judges each of the 252 objects of the full pool, with identical prompts, decoding and seeding, for 1,384 calls with at most two parse failures per judge. Same-version τ on the pool is 0.703 and 0.684 for old and new Qwen, 0.699 and 0.815 for Gemma, and 0.343 and 0.763 for Mistral. Cross-version τ lies below both versions' own values in each case: 0.613 for Qwen and 0.506 for Gemma on the pools of Table 2, and 0.192 for Mistral on the 252 objects of the full pool, a different population from the 220 objects behind the 0.175 of Table 2. The old Mistral is itself a noisy instrument, so part of its cross-version change is sampling noise.

Table 2: Endpoints of the three replacements, from the pool analysis saved at collection time. Positive Δ_{ver} means the new judge is stricter, and intervals are 95% bootstrap intervals. Pool statistics use 218, 217 and 220 objects, and separation uses the anchors. For Qwen, the decision comparisons of Table 1 instead use pool scores completed in Transformers.

	Qwen 3.5→3.6	Gemma 3→4	Mistral 3.1→3.2
Anchor Δ_{ver}	-0.310	$+0.303$	-0.650
95% interval	$[-0.487, -0.146]$	$[+0.014, +0.607]$	$[-0.961, -0.339]$
Pool Δ_{ver}	-0.332	$+0.110$	-0.450
95% interval	$[-0.432, -0.236]$	$[-0.034, +0.258]$	$[-0.616, -0.290]$
Pool Kendall τ	0.613	0.506	0.175
Tied pairs	18%	26%	86%
Separation, old→new	$1.88 \to 3.17$	$1.00 \to 3.92$	$0.73 \to 0.24$

Decision protocols. The threshold is 6, the acceptance line in Figure 6. An object is newly accepted when its old score is below 6 and its new score is at least 6, and newly rejected in the reverse case; neither judge is treated as ground truth. Top-30% turnover is $1 - |T_o \cap T_n|/k$ with $k = \text{round}(0.3\,n)$, and because of ties we report the interval over every tie-break consistent with the scores. Calibration maps are increasing affine maps from old to new scores fitted on the anchors and evaluated on the pool; isotonic fits yield identical threshold decisions on all three pools.

Qwen pool calls missing from the saved vLLM logs were made in Transformers at the pinned revisions with the frozen prompt, temperature 0.6, top-p 0.95, a 6144-token cap and the scheduled seed. Of 219 scores per judge, 110 old-judge and 98 new-judge scores come from vLLM. Splitting the 218 paired objects by the runtime of each of their two scores gives four slices of 45 to 63 objects; each slice changes 10 to 13 decisions, 9 to 12 of them new acceptances, so the two runtimes behave alike.

Table 1 compares the two policies on one population. Qwen and Gemma have similar correlations and similar numbers of threshold changes, but in opposite directions: Qwen newly accepts 41 proposals and newly rejects 7, whereas Gemma newly rejects all 43. Repeating Gemma 3 changes 1 of 217 decisions and repeating Gemma 4 changes 12, so the cross-version count exceeds both. Repeated Qwen calls change 11 of 105 and 17 of 91 decisions; these repeat populations differ from the pool, so they serve as descriptive references and cannot support a matched test. Mistral's 11 changes, against 10 and 0 on repeats, are few only because nearly every new score lands above 6. For the ranking policy, Gemma's old pool scores take four values and 137 of 217 objects tie at the top-30% cutoff, so the compatible turnover runs from 12% to 75%; Mistral's new scores put 204 of 220 objects on one value, and its turnover runs from 3% to 100%.

The anchor-fitted affine map for Qwen has slope 0.83 and leaves all 48 pool changes in place. The Gemma map removes 7 of 43 and leaves 36, still above either repeat. The Mistral map removes all 11 by pushing every pool score above 6, which copies the collapsed judge and restores no ranking. All three slopes are positive, so the top-30% interval is unchanged.

Compatibility and external quality. Gemma's 43 newly rejected proposals have mean reviewer soundness 1.933, against 2.693 for proposals both judges accept. Qwen's newly accepted proposals average 2.672, against 2.749 for jointly accepted ones, and Mistral's 11 newly accepted proposals average 2.050, against 2.561. Reviewer soundness is an external label shipped with the objects and is itself imperfect. Two simple decision rules illustrate why the loss must be audited directly. A location-only rule, which recalibrates when the absolute standardized gap is at least 0.2 and otherwise keeps the old threshold, keeps Gemma's scores despite 43 threshold disagreements because its pool gap is only $+0.110$ SD, and it matches Mistral's new decisions by accepting the weaker tail. A correlation rule, which rescores when cross-version τ is below both same-version repeat values, rescores all three pools. The 0.2 cutoff is an analysis choice for this comparison.

Table 3: Open-weight models of the proposal study and the proposal source, with the first twelve hexadecimal digits of each Hugging Face revision. The Gemma repositories are non-gated mirrors of `google/gemma-3-27b-it` (Gemma Team, 2025) and `google/gemma-4-31B-it`. Phi-4 (Abdin et al., 2024b) generated fact packets, defect summaries, masked titles and revisions and never served as a judge.

Role	Repository	Revision
Qwen, old	`Qwen/Qwen3.5-27B`	`fc05daec18b0`
Qwen, new	`Qwen/Qwen3.6-27B`	`6a9e13bd6fc8`
Gemma, old	`unsloth/gemma-3-27b-it`	`7a5a3053dbd5`
Gemma, new	`unsloth/gemma-4-31B-it`	`51c9f6265ea3`
Mistral, old	`mistralai/Mistral-Small-3.1-24B-Instruct-2503`	`68faf511d618`
Mistral, new	`mistralai/Mistral-Small-3.2-24B-Instruct-2506`	`95a6d26c4bfb`
Text generator	`microsoft/phi-4`	`2db69c1c3e91`
Proposals (dataset)	`hosytuyen/SoundnessBench`	`380f235afcae`

B.1 Judges, decoding, parsing and prompts

Decoding. The open-weight judges ran in vLLM with bfloat16 weights. Every call is stateless and uses the model's default chat template, temperature 0.6, top-p 0.95 and the seed CRC32 of the call identifier. The output cap was 4,096 tokens for the first Qwen calls of phase 0a and 6,144 for every later call. Anchors and evidence-phase objects receive three L0 calls and are summarized by the median; held-out pool objects receive one call per pass. Phi-4 decoded at temperature 0 with at most 1,024 tokens for fact packets and defect summaries, and at temperature 0.3 with at most 3,072 tokens for revisions. The gateway judges of the code population received the prompt as a single user message at temperature 0, with an output cap of 4,000 tokens in the first study and 8,000 tokens, rerun at 16,000, in the second. Their labels, exactly as requested, are `gemini-3.5-flash` to `gemini-3.8-flash`, `claude-opus-4-6`, `-4-7`, `-4-8` and `claude-opus-5`, `claude-sonnet-4-6` and `-5`, `claude-fable-5` and `-5-1`, `grok-4.5` and `grok-4.6` in the first study, and `glm-5.1` to `glm-5.3`, `qwen3.5-plus-2026-02-15`, `qwen3.6-plus-2026-04-02`, `qwen3.7-plus`, `MiniMax-M2.5` and `MiniMax-M2.7` in the second. The DeepSeek proposal pair (DeepSeek-AI, 2026) was fixed at temperature 0.6, top-p 0.95 and 2,048 tokens, and the GPT pair used provider defaults (Appendix C.4). The probability readout of Appendix C.2 scores the ten score strings under Phi-3.5-mini (Abdin et al., 2024a), GLM-4-9B (Team GLM, 2024) and Phi-4.

Parsing. A proposal-study reply is read from the last JSON object in it that carries a numeric `score`; the value is rounded and must lie between 1 and 10, and reasoning text before the object is ignored. Any other reply is a parse failure and is never imputed. Gateway replies are read from the last `"score":` integer in the reply, again restricted to 1 to 10, and pairwise replies from the last `"better"` field after any reasoning block.

Proposal prompts. Every proposal is rendered as a Markdown document with the headings Title, Subfield, Hypothesis, Background and Related Work, Abstract and Planned Experiments, each followed by the corresponding field. The L0 prompt has the system message "You are an expert machine learning researcher serving as a reviewer." and the user message

> Below is a frozen machine-learning research proposal.
>
> *proposal*
>
> Task: evaluate this proposal using the following rubric and assign a single overall score on a 1-10 integer scale.
>
> Rubric:
> - Soundness: are the claims, assumptions, and planned analyses methodologically valid? Are decisive baselines and controls included?
> - Significance: would the results, if obtained, matter to the subfield?
> - Novelty: does the proposal go meaningfully beyond existing work described in its background section?
> - Feasibility: can the planned experiments realistically produce the evidence needed to test the hypothesis?

> Scale anchors:
> 1-2 = fatally flawed or vacuous; 3-4 = serious unresolved defects; 5-6 = borderline, meaningful weaknesses; 7-8 = solid with minor weaknesses; 9-10 = exceptionally rigorous and important.
> Respond with a JSON object on the final line of your reply, in exactly this format:
> {"score": <integer 1-10>, "justification": "<at most three sentences>"}

L1 and L1P insert after the proposal "The following verified factual notes about this proposal are provided for your reference. They are purely descriptive and carry no evaluation:" followed by the six real or placebo facts. L2 and L2P add after the facts "Additionally, an independent methodological audit of this proposal has been performed. The audit certificate below reports findings only; it contains no severity assessment and no score, and the final judgment remains entirely yours:" followed by the object's own or a donor's defect summary. The rest of the prompt is unchanged. The defect-count probe B asks the judge to "enumerate the distinct methodological defects you can identify in this proposal on your own", without a score or suggestions, as a JSON list. The recognition probe R asks whether the judge recognizes the proposal as a specific existing work, without guessing, and returns `recognized`, `paper_title`, `outcome_known` and `confidence`; a recognition counts as verified when the returned title, lowercased and stripped of punctuation, contains or is contained in the source title or has a SequenceMatcher ratio above 0.55 with it. The code-review domain uses the same structure with the system message "You are an expert software engineer serving as a code reviewer.", a submission in place of the proposal, and the criteria correctness, robustness, clarity and efficiency with correspondingly worded anchors.

Code prompts. The pointwise gateway prompt is

> You are reviewing a Python function for correctness and soundness.
> Rate the implementation on an integer scale from 1 to 10:
> 10 = certainly correct, handles the specified behaviour and edge cases
> 7 = probably correct, no defect identified
> 4 = questionable, a plausible defect or unhandled case
> 1 = certainly incorrect, a clear defect that produces wrong results
> Judge only the implementation shown. Do not run it. Be consistent across submissions.
> Write at most three sentences of justification, then output ONLY a JSON object on the final line, with no code fences:
> {"score": <integer 1-10>}
> SUBMISSION:
> *the function in a Python code block*

The pairwise prompt begins "You are comparing two candidate implementations of the same Python function. Decide which implementation is more likely to be correct for every input the docstring specifies. Judge only the code shown. Do not run it.", asks for at most two sentences of justification and a final line {"better": "A"} or {"better": "B"}, and then shows IMPLEMENTATION A and IMPLEMENTATION B in code blocks.

C SUPPLEMENTARY DIAGNOSTICS

This appendix condenses the remaining diagnostics on the proposal and code populations. Each bears on the reuse question only indirectly: it shows which properties of a replacement a small audit can reveal and which it cannot. Appendix H lists their call counts and which of them were specified in advance.

C.1 SHARED EVIDENCE AND TITLE MASKING

Two versions can disagree because they know different things or because they respond differently to the same information. We tested this with nested prompt conditions. L1 adds to the L0 prompt atomic facts about the proposal extracted by Phi-4, a model family different from every judge, and L1P replaces them with placebo facts that concern the research area but say nothing specific about the proposal. L2 adds a defect summary derived from the reviews, and L2P a summary taken from a different proposal and matched for length and style. The facts are nearly free of evaluative language

Table 4: Evidence endpoints on the Qwen and Gemma replacements, with one Mistral replication, in standardized units and with 95% bootstrap intervals. The pooled rows first multiply each replacement by the sign of its L0 gap, so a positive value means the added evidence closes more of the gap. $D_{\text{content}} = \Delta_{\text{ver}}(\text{L1P}) - \Delta_{\text{ver}}(\text{L1})$. The defect-summary rows use the 33 Qwen and 22 Gemma flawed objects. The masking row has no interval: only the two group intervals were saved, and the Qwen call logs of that phase are incomplete in the saved records, so it cannot be recomputed.

Endpoint	n	Estimate and interval
Facts against rubric only, pooled $\Delta_{\text{ver}}(\text{L0}) - \Delta_{\text{ver}}(\text{L1})$	128	$+0.060$ $[-0.039, +0.158]$
Content beyond placebo facts, pooled D_{content}	128	-0.088 $[-0.190, +0.012]$
Qwen, own sign	80	$+0.091$ $[-0.056, +0.238]$
Gemma, own sign	48	-0.083 $[-0.206, +0.041]$
Defect summary on flawed objects, pooled	55	$+0.181$ $[-0.014, +0.375]$
Defect summary minus donor summary, paired	55	$+0.214$ $[+0.059, +0.368]$
replication on Mistral	33	-0.082 $[-0.493, +0.329]$
Title masking, difference in differences	62	$+0.095$ —

and match about 5% of reviewer-verified risk factors, whereas the defect summaries match about 29%.

Table 4 gives the results. Real facts do not close the gap more than placebo facts: the content-specific effect is -0.088 SD with a 95% interval of $[-0.190, +0.012]$, while placebo packets alone move the gap toward zero by 0.11 to 0.17 SD. The defect summaries give an inconclusive primary endpoint. An exploratory paired contrast against donor summaries is positive, but its pre-specified replication on Mistral is too imprecise to settle the question. Shown a donor summary that does not match the proposal, the old Qwen lowers clean-object scores by 0.95 points $[0.49, 1.46]$ and the new Qwen by 0.30 $[0.02, 0.60]$, so susceptibility to authoritative but mismatched text can itself change with a replacement.

Some proposals are recognizable as published papers, and a judge that recognizes one may score it by its known outcome. To test this, titles of 62 objects were rewritten by an LLM until they shared no 5-gram with the original and a SequenceMatcher ratio stayed below 0.55; 31 of these objects had been recognized only by the new Qwen and 31 by neither version. Masking reduced the new Qwen's verified recognition from 29% to 3%, and the difference in differences of the gap between the recognized and control groups was $+0.095$ SD. The saved per-group masking effects, $+0.084$ SD $[-0.147, +0.326]$ for the recognized objects and -0.011 SD $[-0.263, +0.242]$ for the controls, both have intervals that include zero. We use the masked titles only as this diagnostic.

The protocols of the proposal study pre-specify eleven scalar endpoints with a test: the three anchor gaps, the pooled fact and content effects, the pooled defect-summary effect and its Mistral replication, the masking difference in differences, and three endpoints of the revision phase. The protocol of the last phase assigns its content effect to the same multiplicity family as the others. A Benjamini-Hochberg correction (Benjamini & Hochberg, 1995) at $q = 0.05$ over the eleven endpoints keeps the Qwen and Mistral anchor shifts and removes Gemma's, whose unadjusted p is 0.045 (Table 5).

C.2 Score interface

Mistral's concentrated scores do not mean that all its outputs are identical. Across 252 pool calls it has no parse failures, ordinary output lengths and object-specific rationales, with mean pairwise 3-gram Jaccard overlap 0.047. Asked to count defects before scoring, it separates clean from flawed anchors by 0.55 SD, compared with -0.27 for its predecessor. The collapse does not depend on temperature: the between-object standard deviation on the anchors is 0.31 to 0.35 at temperatures 0, 0.6 and 1.0, against 0.94 for the predecessor. A semantically equivalent paraphrase of the rubric raises it to 0.45 to 0.55 and separation to at most 1.64, still at least 42% below the predecessor in every cell, while leaving the predecessor nearly unchanged at 0.87. On a code-review task the same weights keep their discrimination (Appendix C.5). The failure therefore belongs to an operating point defined by weights, prompt, domain and interface, which only an anchor audit on that operating point reveals.

Table 5: Benjamini–Hochberg correction at $q = 0.05$ over the eleven scalar pre-specified endpoints of the proposal study. Values marked ^{a} are two-sided normal approximations from the saved estimate and its 95% bootstrap interval or standard error; the others are the saved test results, and the Mistral replication is one-sided. The saved correction covers ten of these endpoints; the content effect D_{content} is added with the same approximation because the protocol of the last phase assigns it to this family.

Endpoint	p	kept
Anchor gap Δ_{ver}, Mistral	0.00004^a	yes
Anchor gap Δ_{ver}, Qwen	0.00036^a	yes
Anchor gap Δ_{ver}, Gemma	0.045^a	no
Defect summary on flawed objects, pooled	0.069	no
Content beyond placebo facts, pooled D_{content}	0.088^a	no
Facts against rubric only, pooled	0.23^a	no
Title masking, difference in differences	0.27^a	no
Revision: sensitivity against evaporation, Spearman ρ	0.34	no
Revision: in-loop minus fresh-seed score	0.51^a	no
Revision: judge-specific minus generic feedback	0.61^a	no
Defect minus donor summary, Mistral replication	0.66	no

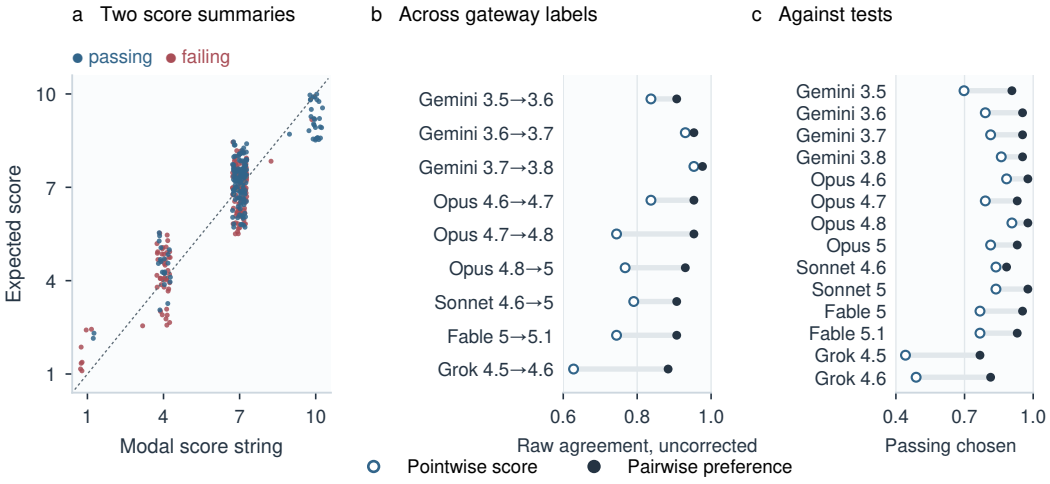


Figure 8: Output protocols resolve different distinctions. Panel a compares the modal score string with the expected score over the ten candidate strings for Phi-4, a single-model probe without sampling. Panel b reports raw cross-version agreement across gateway labels, uncorrected for chance; Cohen's κ favors pairwise choices in only 5 of 9 comparisons. Panel c reports the share of passing programs selected on the same 43 matched pairs. Pointwise decisions compare separately elicited median scores, whereas pairwise prompts present both programs and require a choice. Ties count as no choice in panel c, and matching ties count as agreement in panel b.

Two other interfaces are shown in Figure 8. Following the probability-weighted readout of Liu et al. (2023), three open-weight models score the 356 code objects by the normalized likelihoods of the ten score strings, without sampling any text. Modal scores tie 0.28, 0.51 and 0.54 of object pairs, expectations have no exact ties, and top-30% precision against the execution label rises from 0.61 to 0.68, 0.55 to 0.59 and 0.63 to 0.66. These judges are single models, so the probe does not test stability across versions. In the first gateway study, the fourteen labels also compared the 43 passing-failing anchor pairs in both orders, twice each. Left-side choice rates of 0.49 to 0.58 and order consistency of 0.81 to 0.98 constrain, but do not exclude, position effects. Pointwise judgments tie 18% of pairs on average and choose the passing program on 0.44 to 0.91 of pairs, against 0.77 to 0.98 for pairwise choices. Raw cross-version agreement averages 0.80 pointwise

Table 6: Explained fraction of the two-parameter recalibration. The first row of each block is the pre-specified evaluation; the other rows are post hoc. Rows with a single value are one fit; pool-subset rows give the median, interquartile range and share of 200 draws that meet the two-thirds criterion.

	Fit set → evaluation set	explained	interquartile range	pass
Qwen	24 anchors → 56 evidence-phase objects	-0.15		no
	24 anchors → 218 pool objects	0.89		yes
	24 pool objects → rest of pool	0.64	$[0.43, 0.85]$	47%
	40 pool objects → rest of pool	0.76	$[0.58, 0.87]$	63%
	80 pool objects → rest of pool	0.81	$[0.67, 0.90]$	77%
Gemma	24 anchors → 24 evidence-phase objects	-12.50		no
	24 anchors → 217 pool objects	-1.93		no
	24 pool objects → rest of pool	-0.57	$[-1.43, 0.38]$	12%
	40 pool objects → rest of pool	-0.14	$[-0.79, 0.49]$	15%
	80 pool objects → rest of pool	0.25	$[-0.26, 0.63]$	23%

and 0.93 pairwise, higher for pairwise choices in all 9 adjacent comparisons, but chance-corrected agreement favors them in only five of nine. The data show higher raw agreement under this protocol; they do not establish that pairwise judgments are generally more stable across replacements.

C.3 Recalibration

A two-parameter map from old to new scores was pre-specified to pass if it explained at least two thirds of the gap, measured by $1 - |\text{corrected gap}|/|\text{raw gap}|$. The specification fits the map on the 24 matched anchors, the 12 clean and 12 flawed proposals, and evaluates it on the other objects of the evidence phase, which include the other 24 anchors. This pre-specified evaluation fails on both replacements (Table 6). The pool rows were added afterwards and are post hoc. Evaluated on the 217 or 218 pool objects, the same anchor fit explains most of Qwen's gap but not Gemma's. The Qwen fit has slope 0.99, so it is essentially an upward shift of about 0.6 points. It closes 89% of the mean pool gap, yet at threshold 6 it changes 0 of the pool decisions of Table 1, because the shift is too small to carry an integer score across the threshold, and all 48 disagreements remain. Removing Qwen's location difference thus leaves its decision loss where it was, as the 48-anchor map of Appendix B also shows. The Gemma failure comes from the anchors: the old Gemma compresses them into scores between 6 and 8, with a standard deviation of 0.70, which leaves the slope unidentified and produces an extrapolated slope of 2.5 to 2.7. Refitting on random pool subsets, with 200 draws per size, recovers much of Qwen's mean gap and little of Gemma's. Even a linear map fitted on the whole held-out pool absorbs less than 8% of the object-level dispersion of the gap, with a residual standard deviation of 1.06 against a raw 1.14 on Qwen and 1.29 against 1.33 on Gemma.

When some pool objects are rescored, the rescored objects can also train a repair. On Qwen, anchor-based isotonic calibration raises F1 against the new judge's top-30% set from 0.777 to 0.833 and cuts the regret in mean reviewer soundness by 40%. With 80 to 150 rescored pool objects, a posterior method that uses the rescored objects reaches F1 of 0.901 to 0.944, while the isotonic map keeps the lower regret, 0.32 against 0.46. Matching the new judge and selecting better proposals are different objectives.

C.4 Gateway comparisons on the proposals

Two further replacements were run on the proposal anchors through an API gateway, with the verbatim L0 prompt and three calls per object, 288 calls each and no parse failures. For the GPT labels `gpt-5.4-2026-03-05`→`gpt-5.5-2026-04-23`, the endpoint rejected sampling parameters, so both judges decoded with provider defaults under a rule fixed before the first call. The anchor gap is $+0.452$ SD $[+0.319, +0.572]$, the opposite sign to the two open-weight shifts, separation moves from 3.52 to 3.11, Kendall τ is 0.80, and threshold disagreement is 12.5%. Both gateway configurations then scored the 220 pool objects, the new one twice, for 660 calls. The pool gap is larger, $+0.735$ SD $[+0.641, +0.824]$, and pool disagreement is 32.7%, so the anchor audit understated this shift. Isotonic calibration fitted on the anchors reduces pool disagreement from 32.7% to

19.1% while leaving the top-30% set unchanged under the fixed tie rule. The new configuration's two passes give a same-version τ of 0.824 and turnover of 18.2%, against a cross-version τ of 0.661 and turnover of 27.3%. For the DeepSeek service tiers `v4-flash`→`v4-pro`, the anchor gap is -0.42 SD $[-0.60, -0.25]$, separation improves from 1.78 to 2.12, τ is 0.73 and disagreement is 16.7%. On the anchors, single-call same-version τ exceeds the corresponding cross-version value for every replacement: 0.78 and 0.72 against 0.65 for Qwen, 0.62 and 0.83 against 0.47 for Gemma, 0.37 and 0.55 against 0.23 for Mistral, and 0.89 and 0.92 against 0.85 for the GPT pair.

C.5 Other checks

Residual structure. Using the second pool pass as an independent replicate, the split-half reliability of the per-object gap is 0.25 for Qwen, 0.81 for Gemma and 0.33 for Mistral. On Gemma, object features such as year, subfield, reviewer scores, length and recognition explain $R^2 = 0.46$ of the single-pass gap, permutation $p < 5 \times 10^{-4}$; the newer Gemma rates longer, more rigorous and recognized proposals relatively more favorably. Rescoring the 24 pool objects with the largest Gemma gap after compressing them to about 60% of their length, or after removing hedging, changed the absolute gap by $+0.02$ SD $[-0.15, +0.21]$ and $+0.04$ SD $[-0.06, +0.15]$; the texts were edited by Phi-4 before any judge call, and the unedited arm reproduced the pool gap with $r = 0.93$. Length and style therefore predict which objects move without causing the movement.

Replacement across families. All six open-weight judges scored the same 220 pool objects, so every pair from different families can be read as a replacement. The twelve cross-family pairs give τ between 0.18 and 0.56, top-30% turnover between 40% and 77% and threshold disagreement of up to 59%, against τ between 0.19 and 0.61 for the three within-family replacements. The new Gemma also tracks reviewer soundness best of all six judges, Spearman $\rho = 0.59$ against 0.29 to 0.35, so a replacement can be the best available instrument and a break in the ranking at the same time.

Rationale coding. A frozen coder, DeepSeek `v4-flash`, coded the paired old and new rationales for 188 to 217 pool objects per replacement. The coded change in stance tracks the change in score with Spearman ρ between 0.62 and 0.69. Among objects whose score moved by at least one point, the share with the same substance and a different score is 46% for Qwen, 19% for Gemma and 16% for Mistral, so on two of the three replacements most movement comes with a change in the defects cited. The coder is a single model without human calibration.

Anchor sample size. Taking the full-anchor diagnosis as reference, stratified subsamples of 16 to 24 anchors reproduce it in at least 94% of draws on Qwen and Gemma. Unstratified samples of 8 anchors contain no clean and flawed pair in 62% of draws, so stratification is necessary. These resamples supply no guarantee for other pools or policies.

Revision against one judge. Forty proposals went through six rounds of revision by a third-family model using one judge's feedback, and 20 received generic revision instead. Revision raised fresh-seed scores by $+0.304$ SD $[+0.096, +0.527]$, but judge-specific feedback exceeded generic revision on held-out judges by only $+0.062$ SD $[-0.174, +0.302]$.

Code review. All six open-weight judges scored 48 code-review objects built from HumanEvalPack (Muennighoff et al., 2024), 24 canonical solutions and 24 with an injected, documented bug, three times each under a code-review version of the rubric, for 864 calls. Qwen's gap is -0.08 SD $[-0.17, +0.01]$, with separation of 4.9 and 4.6 under the two versions. Gemma's sharper discrimination replicates, from 1.61 to 4.64 SD. Mistral shifts strongly toward leniency, -0.79 SD $[-0.95, -0.64]$, but does not collapse: its modal share is 31%, against 93% on the proposal pool. The failures are not specific to proposals, but their sign and size depend on the domain, and an anchor audit speaks only for the domain its anchors come from.

D Finite-population audit: proof and implementation checks

For the true D, the discrete cumulative probability $F_D(K) = \Pr_D(K' \leq K)$ is super-uniform: $\Pr_D\{F_D(K) < \alpha\} \leq \alpha$. If $D > D_U(K)$, the true D is excluded from the feasible set in Equation 2,

so $F_D(K) < \alpha$. Thus $\Pr_D\{D > D_U(K)\} \leq \alpha$. Subtracting the observed K and dividing by $M - n$ proves Proposition 2. An erroneous approval implies $R_{\text{rem}} > U(K)$, which proves the second statement. Discreteness may make the bound conservative, and no asymptotic approximation or independence between objects' labels is needed because the labels are fixed and the sample is randomized.

The audit implementation records input hashes, object assignments, fitted thresholds, bounds, remaining disagreement and external correctness. Exact enumeration over every possible D for $(M, n) = (12, 4), (30, 12), (100, 48)$ checks the implementation, with a largest noncoverage probability of 0.04963. The wider enumeration of Appendix E, over every pool size up to 25, reaches 0.049774; both grids stay below $\alpha = .05$, and the values differ only because the grids differ. Equal-call controls compare old repeat 0 against new repeat 0 and each label's repeat 0 against repeat 1 on the identical score-blind task cohort; they describe repeat variation and enter no audit sample. Incomplete objects are excluded before random splitting, so the guarantee concerns the complete-case population.

For prospective use, let Z denote the vector of potential disagreement labels, fixed conditional on any disjoint fitting data. If the audit is a uniform sample conditional on Z, the bound applies to every realization, and the law of total probability gives $\Pr_{Z,S}(\text{approve}, R_{\text{rem}} > \varepsilon) \leq \alpha$. Observing some scores therefore does not require buying all potential scores in advance. The target is the set of decisions defined by that realization, and another independent draw of judge outputs would define a new target. Adaptive routing, sample-dependent scoring, fresh target draws or selection of the candidate after the audit require a different justification.

E CERTIFIABILITY FRONTIER: DERIVATION AND WORKED EXAMPLE

Table 7: First audit size that can approve with zero disagreements at $\alpha = .05$. The pool size M excludes fitting objects. These are exact design calculations. Feasibility at larger n must still be checked because the remaining error allowance is discrete.

M	$\varepsilon = 1\%$	5%	10%	20%
50	48	48	39	15
95	91	74	33	15
250	238	64	30	14
1000	348	60	29	14
10000	300	59	29	14

Table 8: A fixed audit with $M = 95$, $n = 48$ and $\alpha = .05$. The sample rate and the confidence bound answer different questions. The last column applies a 10% tolerance to the 47 unaudited objects.

Observed k	Sample rate	$D_U(k)$	$U(k)$	Approve?
0	0.0%	4	8.5%	Yes
1	2.1%	7	12.8%	No
2	4.2%	9	14.9%	No
3	6.3%	12	19.1%	No

Write $m = M - n$ and $b = \lfloor \varepsilon m \rfloor$ for $0 \leq \varepsilon < 1$. For fixed k, the hypergeometric lower tail $F_D(k)$ is nonincreasing in D, so the retained values of D in Equation 2 form an integer prefix beginning at k. The bound permits at most b remaining errors exactly when $k + b + 1$ is excluded, which proves Equation 4. Because $b < m$, this first excluded candidate lies inside the feasible support, and equality $F_{k+b+1}(k) = \alpha$ retains it and does not approve.

Approval counts also form a prefix in k. Under total count $D = k + b + 1$, the event $K \leq k$ is equivalent to at least $b + 1$ errors among the m unaudited objects. Its probability increases with D, as a coupling that adds one error at a time while keeping the random sample fixed shows. Thus $F_{k+b+1}(k)$ is nondecreasing in k with M, n, b held fixed, which proves the $K \leq K_{\max}$ characterization and shows that $K = 0$ is the best case.

At $K = 0$, the boundary population has $b + 1$ errors, and observing zero means that all of them lie outside the audit, which has probability

$$F_{b+1}(0) = \frac{\binom{M-b-1}{n}}{\binom{M}{n}} = \frac{\binom{M-n}{b+1}}{\binom{M}{b+1}}.$$

Testing whether this probability is strictly below α gives Equation 6. If no $n < M$ passes, certification from a proper subset is infeasible under this rule, and a census $n = M$ leaves no scores to reuse. At zero tolerance the best-case criterion simplifies to $(M - n)/M < \alpha$, which no proper audit satisfies for $M \leq 20$ and $\alpha = .05$.

The implementation enumerates every $M = 2, \ldots, 25$, every proper audit size and every feasible observed count. It compares numerical inversion with exact rational hypergeometric probabilities, checks $11{,}600$ count and tolerance combinations, and enumerates noncoverage over all population error counts, with a largest value of 0.049774. Near equality with α, exact integer arithmetic resolves the comparison.

F RETROSPECTIVE CERTIFICATION, COSTS, AND CORRECTNESS

Table 9: Call accounting at $n = 48$, with three calls per new-judge score. Each row summarizes the same 5,400 dependent resplits. Full rescoring averages 353.7 calls. C and B denote continuation and full-batch fallback. Loss is mean disagreement among approved remaining pools; savings average over all resplits.

Policy	ε	Approved	Loss	Calls C	Saved C	Saved B
Unchanged, 24 reserved	5%	0	n/a	353.7	0.0	-216.0
Unchanged, 24 reserved	10%	209	4.8%	348.2	5.5	-202.2
Unchanged, 24 reserved	20%	2177	6.7%	297.1	56.6	-72.3
Fitted, 24 fit	5%	0	n/a	353.7	0.0	-216.0
Fitted, 24 fit	10%	165	5.3%	349.4	4.3	-205.1
Fitted, 24 fit	20%	1883	7.0%	304.7	49.0	-91.7
Unchanged, no fit	5%	0	n/a	353.7	0.0	-144.0
Unchanged, no fit	10%	235	4.6%	344.4	9.3	-128.5
Unchanged, no fit	20%	2489	6.9%	255.9	97.7	20.1

All analyses in this appendix condition on saved scores. The certifiability and cost analysis was written down in a dated and hashed analysis plan before its computations were run, but after the judgments and a first finite-pool audit of the saved resplits existed, so it is retrospective. Its script reads the record files without modifying them, reproduces all 10,800 rows of that audit, including object assignments, fitted thresholds, disagreement counts, bounds and correctness, and records the hashes of its inputs and plan.

Each comparison uses the same 300 seeded permutations as that audit. For the two protocols with 24 reserved objects, fitting uses the first 24 tasks and the audit the next n; the unchanged protocol without fitting audits the first n tasks. The remaining suffix evaluates disagreement. The audit sizes are 16, 24, 32, 40, 48, 64 and 80, with tolerances of 1%, 5%, 10% and 20%. Each size and tolerance defines its own fixed protocol. Choosing one after looking at the retrospective outcomes, or stopping at the first that approves, does not inherit the individual guarantee. At 1% and 5% tolerance the number of approvals is 0 for every protocol and budget, so Tables 12 and 13 report the full grid only at 10% and 20%.

Table 10 compares the bound with the plug-in sample-rate rule, using 24 fitting or reserved objects and $n = 48$. The sample-rate rule increases approval coverage without the erroneous-approval guarantee: at 10% tolerance, its approvals of the unchanged policy include 898 pools above tolerance among $2{,}941$, whereas the bound permits 209 approvals with 19 above tolerance. The theorem bounds the joint probability of approving and exceeding tolerance over all audit draws, and the error fraction among approvals can be larger.

With equal call cost per object, each approved split under continuation saves exactly $3m$ calls and each rejected split saves none, while full-batch fallback adds $3N$ calls on rejection. The tables

Table 10: Retrospective finite-pool audit on one implementation per task. Each split fits on 24 tasks, audits 48, and evaluates all remaining tasks. Counts pool dependent resplits and transitions; they are descriptive, not independent trials. Above tolerance counts approved splits exceeding the stated tolerance; mean loss averages remaining-pool disagreement over approved splits. The bound controls erroneous approval per fixed protocol, not the error fraction among approvals.

Policy	Rule	Tolerance	Approved	Above tolerance	Mean loss
Unchanged	95% bound	10%	209/5400	19	4.8%
Unchanged	Sample rate	10%	2941/5400	898	7.3%
Unchanged	95% bound	20%	2177/5400	14	6.7%
Unchanged	Sample rate	20%	4852/5400	257	8.9%
Threshold fit	95% bound	10%	165/5400	19	5.3%
Threshold fit	Sample rate	10%	2578/5400	861	7.7%
Threshold fit	95% bound	20%	1883/5400	22	7.0%
Threshold fit	Sample rate	20%	4721/5400	289	9.7%

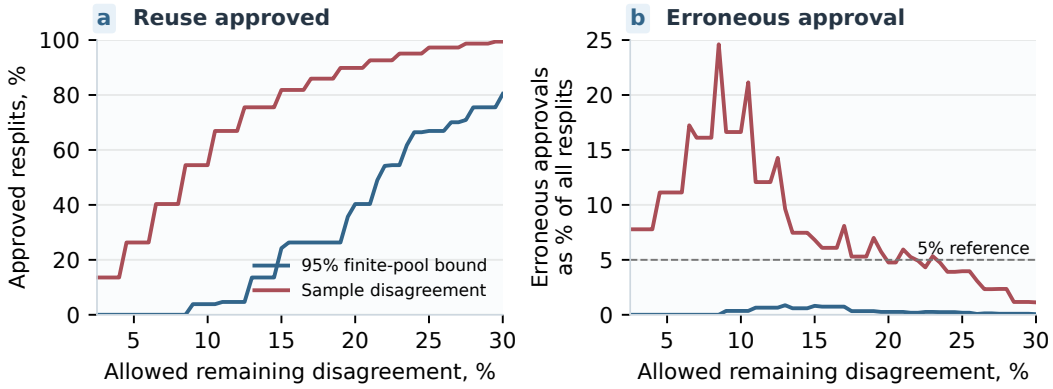


Figure 9: Retrospective coverage comparison for unchanged reuse with 24 reserved tasks and 48 audit tasks. The erroneous-approval denominator includes all 5,400 dependent resplits. Each tolerance is a separate fixed policy, and the curve carries no guarantee that holds across tolerances simultaneously.

report arithmetic averages over the dependent resplits; they do not estimate a distribution of future migrations and carry no confidence intervals that would treat resplits as independent. At $n = 48$, the maximum possible continuation saving is 141 calls for either protocol with 24 reserved objects and 213 without fitting. Both maxima occur in the saved resplits, but low approval coverage keeps the average savings much smaller.

For reuse without fitting at $n = 48$ and 10% tolerance, 10 comparisons never approve in the 300 saved permutations, and mean continuation savings range from zero to 22.9%. The largest value belongs to the Gemini 3.7-to-3.8 comparison, and the preceding Gemini step saves 11.3%. These are deployed configurations evaluated on the same code population. Table 11 reports every comparison at $n = 48$ for both tolerances at which any protocol approves.

Correctness uses the unit-test pass indicator q_i, which is distinct from agreement with the new policy. On an approved remaining pool, the reported accuracy gap is

$$\frac{1}{m} \sum_{i \in \text{rem}} \left[\mathbf{1}\{\Pi_o(s_{o,i}) = q_i\} - \mathbf{1}\{\Pi_n(s_{n,i}) = q_i\} \right].$$

In a complete deployed pool, fitting and audited objects always use new-judge decisions, and after a failed audit every object does. The gap of the hybrid pipeline relative to full rescoring is therefore the same sum multiplied by A/N instead of divided by m. For reuse without fitting at $n = 48$ and 10% tolerance, the mean approved gap is -0.25 percentage points, whereas the gap of the full pipeline averaged over all resplits is about -0.0065 points. The loss column conditions on approval and is not a new bound.

Table 11: Every gateway comparison at $n = 48$: approvals among the 300 resplits and mean continuation savings as a percentage of full rescoring. Reserved and fitted protocols set aside 24 objects for fitting; the protocol without fitting reserves none. The same computation at every other budget and tolerance is part of the reproduction scripts. Names are shortened gateway labels (Appendix B.1), not verified provider snapshots, and the 4 comparisons from the first to the last release of a series overlap their adjacent steps.

	10% tolerance						20% tolerance	
	unchanged, reserved		fitted		no fitting		no fitting	
Comparison	appr.	saved	appr.	saved	appr.	saved	appr.	saved
MiniMax M2.5→MiniMax M2.7	0	0.0	0	0.0	0	0.0	37	7.4
Fable 5→Fable 5.1	16	2.1	13	1.7	25	5.0	273	54.3
Opus 4.6→Opus 4.7	0	0.0	0	0.0	0	0.0	148	29.3
Opus 4.6→Opus 5	2	0.3	2	0.3	0	0.0	187	37.0
Opus 4.7→Opus 4.8	2	0.3	1	0.1	0	0.0	147	29.1
Opus 4.8→Opus 5	6	0.8	3	0.4	4	0.8	245	48.4
Sonnet 4.6→Sonnet 5	21	2.8	19	2.5	25	5.0	279	55.5
Gemini 3.5 Flash→Gemini 3.6 Flash	2	0.3	3	0.4	3	0.6	146	29.0
Gemini 3.5 Flash→Gemini 3.8 Flash	1	0.1	2	0.3	2	0.4	117	23.3
Gemini 3.6 Flash→Gemini 3.7 Flash	52	6.8	23	3.0	57	11.3	300	59.7
Gemini 3.7 Flash→Gemini 3.8 Flash	103	13.6	97	12.8	115	22.9	300	59.7
GLM 5.1→GLM 5.2	0	0.0	0	0.0	0	0.0	1	0.2
GLM 5.1→GLM 5.3	0	0.0	0	0.0	0	0.0	0	0.0
GLM 5.2→GLM 5.3	0	0.0	0	0.0	0	0.0	6	1.1
Grok 4.5→Grok 4.6	0	0.0	0	0.0	0	0.0	5	1.0
Qwen-Plus 3.5→Qwen-Plus 3.6	0	0.0	0	0.0	0	0.0	37	7.4
Qwen-Plus 3.5→Qwen-Plus 3.7	2	0.3	1	0.1	0	0.0	82	16.3
Qwen-Plus 3.6→Qwen-Plus 3.7	2	0.3	1	0.1	4	0.8	179	35.6

Full-batch fallback is a cost scenario that holds the decision reference fixed; fresh calls could change scores, and their correctness and compatibility would need new analysis. Multiplying call counts by another replicate count would not show that fewer replicates keep the decision quality of a median of three. The calculations count new-judge calls and omit token prices, failures, latency and orchestration overhead, where an overhead h reduces the reported saving by h in equivalent call units.

G Release series and a diagnostic rule

A pipeline that upgrades its judge at every release accumulates replacements, and one that skips releases compares scores across several at once. The first gateway study measures both on the code population of Appendix B. Fourteen gateway labels in 5 families scored each object three times at temperature 0 under one rubric, for $14{,}952$ unique calls (Appendix H explains why the record files hold $15{,}786$ lines), and they include two four-release series, Gemini 3.5→3.6→3.7→3.8 Flash and Claude Opus 4.6→4.7→4.8→5. Table 14 and Figure 10 describe every comparison on all 356 objects, and Table 15 applies the diagnostic rule to the anchors and, separately, to the held-out pool.

Diagnostic rule. The rule checks a comparison on the anchors against the version that produced the historical scores, in three steps. A discrimination flag is raised when clean and flawed anchors separate by less than 0.30 SD or when more than half of all object pairs are tied under the new version; an ordering flag when cross-version τ is below the lower same-version repeat τ; and a boundary flag when more than 10% of acceptance decisions differ. The flags suggest withholding, rescoring and calibrating, respectively. The cutoffs are operational; unlike Section 4, the rule carries no error guarantee. The pre-specified rule measured ties by a modal-score share above one half. Applied to all 356 objects, it withheld two Gemini comparisons whose new versions separate clean from flawed inputs by 2.50 and 2.77 SD, so the analysis uses the tied-pair criterion, a post hoc variant that changes no proposal verdict; Table 14 reports both. The repeat reference compares single calls, whereas cross-version scores are medians of three, so the ordering check is an operational reference rather than a noise comparison at equal budget.

Table 12: Complete cost grid at 10% tolerance. I0 is unchanged with no fit; I24 is unchanged with 24 reserved objects; F24 fits on 24. Each row has 5,400 dependent resplits. C/B are mean continuation/full-batch calls; Max is maximum observed continuation savings. Loss and accuracy gap condition on approved remaining pools. The latter is in percentage points.

Policy	n	Approved	Calls C/B	Max	Loss, %	Acc. gap	Above tol.
I0	16	0	353.7/401.7	0	n/a	n/a	0
I0	24	0	353.7/425.7	0	n/a	n/a	0
I0	32	0	353.7/449.7	0	n/a	n/a	0
I0	40	347	338.4/450.7	237	4.78	-0.50	41
I0	48	235	344.4/482.1	213	4.57	-0.25	13
I0	64	443	340.1/516.4	165	4.10	-0.50	11
I0	80	267	347.9/576.0	117	4.29	+0.43	23
I24	16	0	353.7/473.7	0	n/a	n/a	0
I24	24	0	353.7/497.7	0	n/a	n/a	0
I24	32	46	352.1/518.7	186	10.69	-0.18	22
I24	40	308	344.3/525.3	165	4.70	-0.41	25
I24	48	209	348.2/555.9	141	4.80	-0.12	19
I24	64	434	346.2/589.0	93	4.35	-0.50	21
I24	80	38	353.4/663.2	45	7.02	+1.75	10
F24	16	0	353.7/473.7	0	n/a	n/a	0
F24	24	0	353.7/497.7	0	n/a	n/a	0
F24	32	31	352.6/519.6	186	11.50	-0.26	21
F24	40	263	345.6/528.3	165	4.99	-1.02	24
F24	48	165	349.4/558.8	141	5.27	-0.93	19
F24	64	349	347.7/594.6	93	4.36	-0.69	18
F24	80	26	353.4/663.9	45	8.72	-0.51	7

Table 13: Complete cost grid at 20% tolerance. I0 is unchanged with no fit; I24 is unchanged with 24 reserved objects; F24 fits on 24. Each row has 5,400 dependent resplits. C/B are mean continuation/full-batch calls; Max is maximum observed continuation savings. Loss and accuracy gap condition on approved remaining pools. The latter is in percentage points.

Policy	n	Approved	Calls C/B	Max	Loss, %	Acc. gap	Above tol.
I0	16	1434	271.8/307.1	309	6.69	-0.80	3
I0	24	2112	242.5/286.4	285	6.86	-0.85	8
I0	32	2506	233.0/284.4	261	7.01	-0.79	9
I0	40	1918	269.7/347.1	237	6.49	-0.77	13
I0	48	2489	255.9/333.5	213	6.88	-0.67	12
I0	64	2631	273.6/372.1	165	6.99	-0.67	4
I0	80	2370	302.6/437.3	117	6.99	-0.76	35
I24	16	1331	295.5/385.9	237	6.77	-0.76	13
I24	24	1955	276.8/368.7	213	6.63	-0.66	9
I24	32	1366	306.0/431.5	189	6.04	-0.71	6
I24	40	1819	298.3/425.6	165	6.30	-0.70	6
I24	48	2177	297.1/426.0	141	6.72	-0.70	14
I24	64	2645	308.4/443.1	93	7.16	-0.69	34
I24	80	1878	338.0/541.5	45	6.17	-0.93	54
F24	16	1193	301.5/395.0	237	7.18	-1.39	16
F24	24	1782	283.6/380.1	213	7.22	-1.34	16
F24	32	1207	311.5/442.0	189	6.38	-1.32	8
F24	40	1584	305.4/441.1	165	6.59	-1.27	8
F24	48	1883	304.7/445.4	141	6.99	-1.28	22
F24	64	2270	314.8/467.8	93	7.41	-1.24	36
F24	80	1676	339.7/554.9	45	6.39	-1.55	46

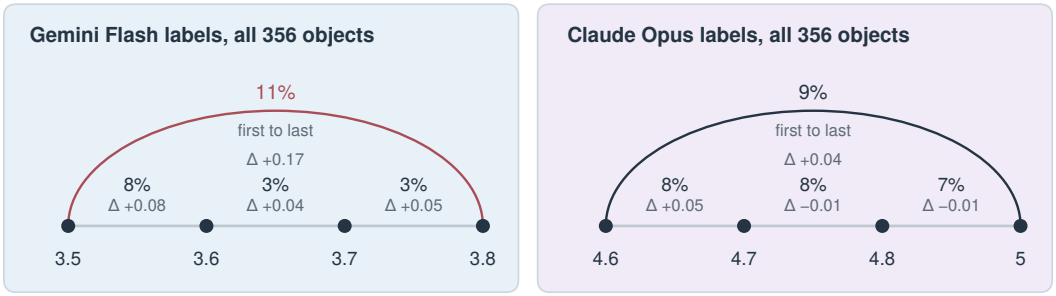


Figure 10: Two gateway release series on all 356 code objects. Edges compare adjacent labels and arcs compare the first and last labels. Each edge reports threshold disagreement and the standardized location gap. On this population Gemini's adjacent disagreements stay below 10%, while its first-to-last disagreement exceeds it. The rule itself uses the anchors, on which the first Gemini step already exceeds 10% (Table 15).

Table 14: First gateway study on all 356 code objects (355 when one version lacks a score): adjacent replacements and, for the two release series, the comparison between first and last release. The verdict columns apply the rule to this whole population for description; the rule's own population is the anchors, reported with the held-out pool in Table 15. The pre-specified rule measures ties in the discrimination check by modal-score share, and the revised rule, a post hoc variant, by the share of tied pairs. τ_{rep} is the lower of the two versions' same-version repeat correlations.

						verdict	
Replacement	Δ_{ver}	τ	τ_{rep}	turnover	boundary	pre-specified	revised
Gemini 3.5→3.6	+0.081	0.848	0.901	7%	8%	withhold	rescore
Gemini 3.6→3.7	+0.037	0.916	0.901	7%	3%	withhold	no flag
Gemini 3.7→3.8	+0.054	0.936	0.941	7%	3%	rescore	rescore
Gemini 3.5→3.8	+0.174	0.774	0.930	16%	11%	rescore	rescore
Opus 4.6→4.7	+0.052	0.836	0.773	20%	8%	no flag	no flag
Opus 4.7→4.8	−0.008	0.837	0.784	20%	8%	no flag	no flag
Opus 4.8→5	−0.006	0.858	0.752	16%	7%	no flag	no flag
Opus 4.6→5	+0.037	0.848	0.752	21%	9%	no flag	no flag
Sonnet 4.6→5	−0.007	0.862	0.751	12%	8%	no flag	no flag
Fable 5→5.1	+0.048	0.857	0.724	16%	7%	no flag	no flag
Grok 4.5→4.6	+0.069	0.746	0.700	22%	14%	calibrate	calibrate

Accumulated change. On all 356 objects the sums of standardized steps are close to the gaps between first and last release: +0.172 against a measured +0.174 for Gemini and +0.038 against +0.037 for Opus. On Gemini the first-to-last correlation, 0.774, is lower than that of any step, which range from 0.848 to 0.936, and top-set turnover grows from 7% at the last step to 16% across the series. Gemini's three steps disagree at the acceptance threshold on 8%, 3% and 3% of the 356 objects, each below 10%, while the first and last release disagree on 11% (Figure 10). Location steps in the same direction add up, so disagreement with the release that produced the scores can exceed every step-to-step value.

The anchors show the same ordering of magnitudes, 10.5%, 3.5% and 4.7% for the steps against 14.0% from first to last, but there the first step already crosses the boundary flag and also raises an ordering flag (Table 15). A pipeline applying the rule to each Gemini release would therefore have stopped at the first step; the case in which step-by-step checks stay below the boundary flag while the comparison with the source exceeds it occurs only on the all-object population. On the anchors, the Opus series is flagged at two steps, for ordering at 4.7→4.8 and at the boundary at 4.8→5 (12.8%), and from first to last (11.6%), whereas the pool flags no Opus comparison. Against Gemini 3.5, Gemini 3.8 fails the ordering check on the anchors, the pool and all objects, so a pipeline using both ranks and thresholds would need to assess both relations. Of the three single replacements, only Grok 4.5→4.6 raises the boundary flag on all objects, at 14%; on the anchors, Fable 5→5.1

raises it too, at 14.0% against 4.8% on the pool. In total the anchors raise 6 boundary flags in the 11 comparisons and the pool 1, and anchor and pool verdicts agree in 6.

Table 15: First gateway study under the revised rule of Appendix G, applied to the 86 anchors, the population the rule specifies, and separately to the 270 held-out pool objects, using objects scored by both versions (all 86 anchors and 269 or 270 pool objects). τ is the cross-version Kendall correlation, τ_{rep} the lower of the two versions' repeat correlations on the objects of that row, and boundary the disagreement at the acceptance threshold, given to one decimal because several anchor values lie near the 10% flag. Shaded rows compare the first and last release of a series.

	anchors				held-out pool			
Replacement	τ	τ_{rep}	boundary	verdict	τ	τ_{rep}	boundary	verdict
Gemini 3.5→3.6	0.803	0.867	10.5%	rescore	0.862	0.913	7.8%	rescore
Gemini 3.6→3.7	0.916	0.867	3.5%	no flag	0.917	0.913	2.6%	no flag
Gemini 3.7→3.8	0.939	0.946	4.7%	rescore	0.935	0.932	2.6%	no flag
Gemini 3.5→3.8	0.733	0.942	14.0%	rescore	0.787	0.926	10.0%	rescore
Opus 4.6→4.7	0.806	0.774	8.1%	no flag	0.852	0.766	8.6%	no flag
Opus 4.7→4.8	0.764	0.779	9.3%	rescore	0.858	0.783	8.2%	no flag
Opus 4.8→5	0.785	0.731	12.8%	calibrate	0.875	0.753	5.6%	no flag
Opus 4.6→5	0.842	0.731	11.6%	calibrate	0.848	0.753	8.1%	no flag
Sonnet 4.6→5	0.861	0.715	5.8%	no flag	0.865	0.761	8.5%	no flag
Fable 5→5.1	0.776	0.687	14.0%	calibrate	0.878	0.731	4.8%	no flag
Grok 4.5→4.6	0.734	0.715	12.8%	calibrate	0.748	0.694	14.4%	calibrate

Second gateway study. A second gateway served GLM 5.1→5.2→5.3, Qwen-Plus 3.5→3.6→3.7 and MiniMax M2.5→M2.7, with $8{,}443$ parsed calls; each judge scored the same 356 code objects three times under the same prompt at temperature 0. Candidate pairs were probed with one object beforehand and excluded when the gateway added prompt tokens to one version, when the endpoint failed, or when the judge gave no score within the output budget. Calls truncated at 8,000 output tokens were rerun at 16,000; of 227 truncated calls, 182 then yielded a score, and 56 calls that stopped without a parseable score were kept without rerunning. GLM 5.1 retains 350 objects, GLM 5.3 retains 355 and every other judge all 356. Object scores are medians of the available repeats without imputation, so the verdicts describe the available-score population.

Table 16 applies the revised rule to the anchors and, separately, to the held-out pool. Anchor and pool verdicts agree in 6 of 7 comparisons. All seven anchor comparisons are flagged and the pool flags six; the exception, Qwen-Plus 3.6→3.7, raises ordering and boundary flags on the anchors and none on the pool. There are thus no observed misses, but also no anchor comparison without a flag on which to test permission to reuse, and the pool verdict applies the same rule rather than measuring deployment harm. In the first gateway study, used while the rule was developed, anchor and pool agree in 6 of 11 comparisons (Table 15). An audit sample is also not automatically a calibration sample. An isotonic map fitted on the matched anchors raises pool boundary disagreement in 8 of the 11 comparisons of the first study, including Grok, its only comparison with a pool calibration flag, and it does not lower disagreement for MiniMax, the only such comparison in the second study.

H PROTOCOL, PRE-SPECIFICATION AND SAMPLE ACCOUNTING

Proposal study. The proposal calls were collected in six consecutive phases, labeled 0a, 0b, 1a, 1b, 2 and 3 in the records: anchor scoring (0a), fact packets (0b), defect summaries and title masking (1a), revision (1b), the held-out pool and the Mistral replacement (2), and the Mistral perturbation matrix and placebo extension (3). Before each phase's first judge call, a lock file stored SHA-256 hashes of its configurations, prompts and analysis code. Additions after a phase began were documented before the endpoints they affected were computed, and none changed the definition, sign or significance threshold of an endpoint after it was computed. They include the Gemma replacement and a larger output budget after 18 of 202 old-Qwen calls were truncated (0a), a relaxed acceptance rule for masked titles (1a), 114 missing Gemma baseline calls (1b), the Mistral replication (2), and the two gateway replacements, second pool pass, code-review domain and counterfactual

Table 16: Second gateway study. Each row applies the diagnostic rule of Appendix G to the anchors and, separately, to the held-out pool, using objects scored by both versions: 82 to 86 of the 86 anchors and 267 to 270 of the 270 pool objects. τ_{rep} is the lower of the two versions' repeat correlations, computed on the objects of that row, so one version can contribute different values to different rows: GLM 5.2 gives 0.629 on the anchors it shares with GLM 5.1 and 0.642 on those it shares with GLM 5.3, and the 0.875 of GLM 5.1→5.3 is higher because GLM 5.2, the least repeatable version, is not part of that comparison. Boundary is disagreement at the acceptance threshold. The last two rows compare the first and last release of each three-release series.

	anchors				held-out pool			
Replacement	τ	τ_{rep}	boundary	verdict	τ	τ_{rep}	boundary	verdict
GLM 5.1→5.2	0.518	0.629	23%	rescore	0.579	0.645	24%	rescore
GLM 5.2→5.3	0.519	0.642	23%	rescore	0.595	0.645	19%	rescore
Qwen-Plus 3.5→3.6	0.771	0.865	21%	rescore	0.784	0.850	14%	rescore
Qwen-Plus 3.6→3.7	0.734	0.837	16%	rescore	0.838	0.833	9%	no flag
MiniMax M2.5→M2.7	0.771	0.665	15%	calibrate	0.734	0.598	14%	calibrate
GLM 5.1→5.3	0.481	0.875	32%	rescore	0.497	0.846	24%	rescore
Qwen-Plus 3.5→3.7	0.737	0.837	14%	rescore	0.776	0.833	11%	rescore

Table 17: Sample accounting for the proposal study. Judges: Q is the Qwen pair, G the Gemma pair, M the Mistral pair, A the GPT pair and D the DeepSeek pair. L0 is defined in Appendix B and L1, L1P, L2 and L2P in Appendix C.1; R is a recognition probe and B a defect-count probe. Object sets overlap, since the anchors recur in several phases.

Phase	Object set	Conditions, calls	Judges	Calls
0a	48 anchors	L0 616, B 192, R 192	Q, G	1,000
0b	48 anchors, 96 screening candidates, 32 chosen	L1 768, L1P 768, L0 192, R 192, B 64	Q, G	1,984
1a	80 evidence objects and 62 masked variants	L2 878, L2P 878, L0 372, R 124	Q, G	2,252
1b	40 revision objects with depth and control variants	L0 1,014, B 400	Q, G	1,414
2	48 anchors and the 252-object full pool	L0 1,672, R 1,480, L2 and L2P 396, B 96	Q, G, M	3,644
3	48 anchors, perturbation matrix	L0 672	M	672
GPT anchors	48 anchors	L0 288, three repeats	A	288
GPT pool	220 held-out pool	L0 660, old once, new twice	A	660
Pool repeats	220 held-out (Q, G), 252 full pool (M)	L0 1,384, second pass	Q, G, M	1,384
DeepSeek anchors	48 anchors	L0 288, three repeats	D	288
Code review	48 code-review objects	L0 864, three repeats	Q, G, M	864
Counterfactual edits	24 pool objects in 3 arms	L0 432, three repeats	G	432
Total				14,882

edits, each with endpoints fixed before its first call. The residual-structure, cross-family, rationale and anchor-size analyses reuse frozen calls and are exploratory. Table 17 reconciles every call.

Gateway studies. For the first gateway study, objects, prompts, endpoints, comparisons and the diagnostic rule were fixed before the first call; the analysis code was written after collection began and before endpoints were computed. A first pool of HumanEvalPack programs was replaced after a 252-call smoke test gave only floor and ceiling scores, and the output budget was raised from 1,600 to 4,000 tokens after 21 of 216 Gemini 3.5 calls were truncated; the 260 calls at the old budget were discarded before any retained record was collected. GPT labels were excluded because that gateway prepends a coding-agent preamble of about 4,400 tokens to every request. The tied-pair criterion was adopted after the first study's verdicts were computed. The first study's record files hold $15{,}786$ lines for $14{,}952$ unique label, object and repeat combinations. The difference comes from one resumed shard: the Grok 4.6 file has $1{,}902$ lines for $1{,}068$ calls, with 834 combinations recorded twice, 280 of them with different scores. No other record file of either gateway study, of the pairwise comparisons or of the probability readout repeats a combination. Every analysis keeps the later record in file order. Keeping the earlier record instead moves Grok 4.5→4.6 pool disagreement from 14.4% to 16.3% and pool τ from 0.748 to 0.737; on all objects disagreement moves from 14.0% to 15.4%, anchor disagreement stays at 12.8%, and the verdict remains *calibrate* on every population. For the second gateway study, the configuration file states that the revised rule preceded collection, but its recorded freeze date is later than the observed collection dates, so the logs do not independently establish pre-registration. The truncation reruns followed the initial analysis of

that study, with rule, endpoints and analysis code unchanged. The certifiability and cost analysis is retrospective, as described in Appendix F.