THE VERIFICATION SQUEEZE: FROM VERIFIER SCORES TO DEPLOYMENT DECISIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Standard LLM verifiers are often evaluated by accuracy or AUROC, but higher scores do not show that accepted outputs meet a target reliability level. For such deployment decisions, what matters is *accepted risk*: the fraction of accepted outputs that are wrong, which depends on the solver population's base error. We call the gap between required and supplied evidence the *verification squeeze*. We introduce the Verification Feasibility Audit (VFA), a two-gate framework. Gate 1 asks whether a verifier-selection procedure provides enough evidence for the population and risk target; Gate 2 asks whether independent calibration data support a fixed acceptance rule. Across five LLMs from four providers, 117 matched benchmark–method comparisons, and a 120-condition calibration design, we find three patterns. (1) Reusing evaluation labels during selection can overstate verifier reliability: measured evidence rises by $1.13\times$ at the median, and 3 of 40 controlled conditions cross the evidence threshold. (2) Verifier improvements often do not change the audit outcome: none of the audited-infeasible baselines becomes feasible under the methods we test. (3) Calibration support weakens as the statistical claim is strengthened: seven of eight rules meet the empirical accepted-risk target, five pass the prespecified large-sample screen, while finite-sample methods support none of the applicable rules they evaluate at the available sample sizes. These results show that conventional verifier metrics alone are insufficient for deployment decisions. Verifier evaluations should also report the evaluated population, selection protocol, evidence margin, and supported risk statement.

1 INTRODUCTION

Large language model (LLM) verifiers serve as critical gatekeepers, deciding whether generated answers, plans, or agent actions should be accepted or executed. While verifiers are commonly evaluated by agreement or correlation with human judgments (Zheng et al., 2023; Liu et al., 2023; Kim et al., 2024a; Zhu et al., 2025), these metrics overlook the central operational question: *when a verifier marks an output* PASS*, how often is it actually wrong?* We formalize this quantity as *accepted risk* and investigate it in the reference-free setting, where no gold reference is available at decision time.

Accepted risk depends on the solver's base error rate. The same verifier can therefore imply very different risks across solver populations. If the solver is usually correct, even a moderately accurate verifier can produce a low-risk PASS; if the solver is often wrong, the verifier must separate correct from wrong outputs much more strongly for a PASS to be trustworthy. For example, a verifier with $\mathrm{TPR} = 0.5$ and $\mathrm{FPR} = 0.1$ has an accepted risk of about 2% when the solver error rate is 10%, but about 64% when it is 90%. Consequently, higher balanced accuracy does not by itself imply that a PASS satisfies a chosen risk target. We summarize the evidence supplied by a rule using its positive likelihood ratio, $\mathrm{LR}^{+} = \mathrm{TPR}/\mathrm{FPR}$: *how much more often it accepts a correct output than a wrong one*. Comparing this supply with the evidence required by the population and risk target gives the remaining deficit, which we call the *verification squeeze*.

The *Verification Feasibility Audit* (VFA) turns this deployment question into two checks (Figure 1). Gate 1 asks whether the verifier-selection procedure provides enough held-out evidence for the population and risk target. To avoid evaluating a rule on the same labels used to select it, Gate 1 keeps verifier selection and evaluation separate: the verifier is chosen on one set of data and evaluated on a

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.

disjoint held-out set. Gate 2 then assesses what level of support reserved calibration data provide for a pre-fixed acceptance rule. Together, the two gates ask whether a verifier that looks strong by conventional metrics still has enough evidence, and enough calibration support, to justify a deployment decision.

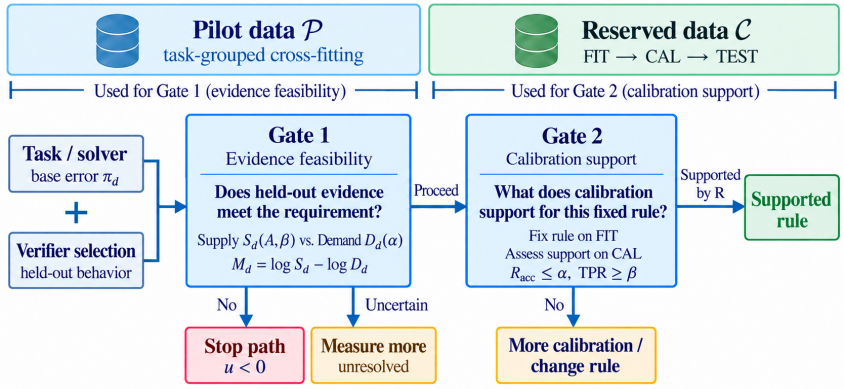


Figure 1: **Overview of the Verification Feasibility Audit (VFA).** Gate 1 uses task-grouped pilot data $\mathcal{P}$ to compare held-out verifier evidence with the evidence required by the population and risk target. Gate 2 uses reserved data $\mathcal{C}$, with separate FIT, CAL, and TEST roles, to determine what calibration supports for one fixed acceptance rule. The resulting action depends on the chosen risk-control procedure and its assumptions.

Across our experiments, three results stand out. First, reusing evaluation labels when choosing a verifier can make it look more reliable than it is: measured evidence increases by a median factor of 1.13×, and 3 of 40 controlled conditions cross the evidence requirement (Section 4.2). Second, better verifier scores do not necessarily make a verifier strong enough for the target risk: none of the audited-infeasible baselines becomes feasible under the verifier improvements we test (Section 4.3). Third, calibration can remain inconclusive even when a fixed rule looks promising: five of eight rules pass a calibration check based on a large-sample approximation, but the finite-sample methods we test support none of the rules they evaluate (Section 4.4).

Together, these results separate three questions that conventional verifier evaluation can blur: *Did the verifier score improve? Is it strong enough for this population and risk target? Do the calibration data support using the chosen rule?* Conventional metrics mainly address the first question. VFA makes the latter two explicit.

Our contributions are:

- We introduce VFA, a two-gate deployment audit that separates selection-aware evidence feasibility from rule-specific calibration support. Gate 1 compares held-out evidence from a verifier-selection procedure with the evidence required by the population and risk target; Gate 2 assesses what independent calibration data support for one fixed acceptance rule (Section 3).
- We make verifier selection part of the evaluation rather than treating the chosen verifier as fixed. Gate 1 measures evidence on data held out from selection, and a controlled label-reuse study shows why this separation matters (Section 4.2).
- We evaluate these ideas at scale across five LLMs from four providers, with 117 matched benchmark–method comparisons and a 120-condition calibration study. The experiments cover routing, repeated judgments, learned aggregation, and cross-vendor panels, and show when verifier improvements provide enough evidence—and when calibration data still do not support the resulting rule (Sections 4.3 and 4.4).

Section 2 develops the connection between verifier metrics, accepted risk, and risk control. Section 3 introduces VFA and its two gates. Section 4 evaluates the selection, scaling, and calibration questions, and Section 5 discusses the implications and limitations.

2 BACKGROUND AND RELATED WORK

Acting on a PASS requires a target accepted risk, an evaluation protocol that accounts for verifier selection, and a calibration procedure whose statement matches the claim made.

2.1 FROM VERIFIER METRICS TO ACCEPTED RISK

Let x denote a solver output and $Y \in \{0, 1\}$ its correctness label, with $Y{=}1$ for a correct output. A verifier method m assigns a score $s_m(x)$, with larger values favouring correctness, and threshold τ defines the acceptance rule $A_{m,\tau}(x) = \mathbf{1}\{s_m(x) \geq \tau\}$, where $A_{m,\tau} = 1$ means PASS. The setting is *reference-free* when the verifier cannot consult a gold answer at decision time; correctness labels remain available offline for selection, calibration, and evaluation (Appendix A).

For a fixed rule, $\mathrm{TPR} = \Pr(A_{m,\tau}{=}1 \mid Y{=}1)$ and $\mathrm{FPR} = \Pr(A_{m,\tau}{=}1 \mid Y{=}0)$. Both condition on correctness and therefore omit how often the solver is wrong. We represent that property of population d by its base error $\pi_d = \Pr(Y{=}0 \mid d)$. The *accepted risk*,

$$R_{\mathrm{acc}} = \Pr(Y{=}0 \mid A_{m,\tau}{=}1) = \frac{\pi_d \,\mathrm{FPR}}{(1-\pi_d)\,\mathrm{TPR} + \pi_d \,\mathrm{FPR}},$$

therefore depends on both verifier rates and the population base error; controlling FPR alone need not meet an accepted-risk target (Appendix B.8).

Accuracy, balanced accuracy, and AUROC summarize related but different properties: overall decision correctness, class-conditional performance, and ranking across thresholds. None alone determines the error rate among accepted outputs on a specified population. Outcome and process verifiers (Cobbe et al., 2021; Lightman et al., 2024) and prompted judges (Zheng et al., 2023; Liu et al., 2023; Kim et al., 2024a; Zhu et al., 2025; Wang et al., 2024) commonly report such prediction or ranking metrics. We instead focus on the reliability of the acceptance decision.

2.2 VERIFIER SELECTION AND SCALING

Developers usually choose among verifier models, prompts, judging directions, aggregation rules, and thresholds. Let $\mathcal{A}$ denote this complete selection procedure. With disjoint samples, $(\widehat{m}, \widehat{\tau}) = \mathcal{A}(\mathcal{D}_{\mathrm{sel}})$ is evaluated on $\mathcal{D}_{\mathrm{eval}}$. Reusing evaluation labels during selection can yield optimistic estimates because the selected configuration reflects sample-specific variation as well as verifier quality; held-out evaluation separates selection from assessment.

Recent evaluator benchmarks differ in whether operating choices use the reported evaluation sample or held-out data (Li et al., 2024; Wang et al., 2025; Liu et al., 2025); Appendix G details these protocols. Test-time verification can scale through repeated judgments (Wang et al., 2023), multi-judge panels (Verga et al., 2024; Lifshitz et al., 2025), alternative scoring or aggregation (Zhao et al., 2026; Kwok et al., 2026), and adaptive compute (Snell et al., 2025), with empirical studies measuring the resulting performance (Zhou et al., 2026). Searching across mechanisms enlarges $\mathcal{A}$ and therefore also requires selection-aware evaluation.

These methods still judge the supplied candidate record. Execution-grounded verification instead adds observations from running the candidate or querying an environment (Chen et al., 2023; Ni et al., 2023; Gou et al., 2024); reference answers, tool results, retrieved state, and human inspection provide other instance-specific evidence. Our focus is whether the out-of-fold evidence delivered by the selected verifier procedure is sufficient for the population-specific accepted-risk requirement.

2.3 SELECTIVE ACCEPTANCE AND RISK CONTROL

A useful acceptance rule must keep errors among accepted outputs low without rejecting nearly everything, as in selective prediction (Geifman & El-Yaniv, 2017). We write α for the accepted-risk target and β for the *retention requirement*, so the objective is $R_{\mathrm{acc}} \leq \alpha$ with $\mathrm{TPR} \geq \beta$. Here β constrains the fraction of correct outputs retained, not overall coverage $\Pr(A_{m,\tau}{=}1)$.

Distribution-free risk-control methods use held-out calibration data to support statements about a fixed rule (Bates et al., 2021; 2023; Angelopoulos et al., 2024; 2025). Different procedures control

different quantities, so marginal coverage, a bound on a monotone loss, and a high-probability upper bound on accepted risk are not interchangeable. Related applications include language-model generation and factuality (Mohri & Hashimoto, 2024; Quach et al., 2024), selective judging and escalation (Jung et al., 2025; Badshah et al., 2026), budgeted evidence acquisition (Xu et al., 2026), and controlled agent decisions (Sadhuka et al., 2025; Akewar & Ranjan, 2026).

VFA is complementary to conformal risk control (Angelopoulos et al., 2024), Learn-then-Test (Angelopoulos et al., 2025), Trust-or-Escalate (Jung et al., 2025), SCOPE (Badshah et al., 2026), and E-valuator (Sadhuka et al., 2025). It asks an earlier question: *does the selected verifier provide enough evidence for this solver population and risk target?* Gate 1 tests this feasibility; Gate 2 then applies a chosen risk-control procedure to a fixed rule.

3 The Verification Feasibility Audit

VFA separates two requirements for selective deployment: *evidence feasibility*, whether the audited verifier-selection procedure provides enough discrimination for the population and targets (α, β); and *calibration support*, whether independent data support using a fixed acceptance rule at the target risk. A full sequential audit reserves task-disjoint pilot and reserved sets, P and C, in advance. Gate 1 uses cross-fitting within P to evaluate evidence feasibility for the search procedure $\mathcal{A}$. If Gate 1 supports proceeding, Gate 2 uses C, with fixed FIT, CAL, and TEST roles, to select and assess one rule without reusing pilot outcomes (Appendices B.4 and B.6).

3.1 Gate 1: Required versus supplied evidence

A PASS provides evidence when it is more likely on a correct output than on a wrong one. The positive likelihood ratio measures this directly: $\mathrm{LR}^+ = \mathrm{TPR}/\mathrm{FPR}$ is how much more often the rule accepts a correct output than a wrong one. Rearranging the Bayes expression for accepted risk turns $R_{\mathrm{acc}} \le \alpha$ into

$$R_{\mathrm{acc}} \le \alpha \quad \Longleftrightarrow \quad \mathrm{LR}^+ \;\ge\; D_d(\alpha) = \underbrace{\frac{\pi_d}{1-\pi_d}}_{\text{base-error odds}} \cdot \underbrace{\frac{1-\alpha}{\alpha}}_{\text{risk target}}. \tag{1}$$

Equation (1) is a direct Bayes rearrangement of the familiar relationship between class-conditional rates, prevalence, and precision (Davis & Goadrich, 2006). We call $D_d(\alpha)$ the *evidence demand*. For the nondegenerate case $0 < \pi_d < 1$, $0 < \alpha < 1$, and $\mathrm{FPR} > 0$, demand increases with the solver's base error and with stricter risk targets.

This relation also gives a population-level feasibility condition. Let $\mathcal{H}_d$ be the finite family of methods and thresholds available to the audited search procedure on population d. Its evidence capacity at retention level β is

$$S_d^\star(\mathcal{H}_d, \beta) = \max_{r \in \mathcal{H}_d:\, \mathrm{TPR}_d(r) \ge \beta} \mathrm{LR}_d^+(r),$$

with population feasibility margin

$$M_d^\star(\mathcal{H}_d, \alpha, \beta) = \log S_d^\star(\mathcal{H}_d, \beta) - \log D_d(\alpha).$$

By Equation (1), $M_d^\star \ge 0$ exactly when the family contains at least one rule with population accepted risk at most α and TPR at least β. Evidence feasibility is therefore necessary for a useful low-risk rule to exist within the audited family.

Because the oracle quantity $S_d^\star$ is not observed, Gate 1 instead evaluates a prespecified search procedure $\mathcal{A}$ on held-out data. Within the pilot set P, each task-grouped fold is held out in turn; on the remaining folds, $\mathcal{A}$ selects a method and threshold subject to $\mathrm{TPR} \ge \beta$, then applies that choice once to the held-out fold. Thus every eligible example receives one out-of-fold decision from a rule chosen without its label. Appendix B.4 gives the fitting and selection roles, threshold candidates, and tie rules.

We pool these out-of-fold decisions before computing rates. Their held-out TPR and FPR define the measured *evidence supply*:

$$S_d(\mathcal{A}, \beta) = \frac{\mathrm{TPR}_{\text{held-out}}}{\mathrm{FPR}_{\text{held-out}}},$$

and the empirical feasibility margin is

$$M_d(\mathcal{A}, \alpha, \beta) = \log S_d(\mathcal{A}, \beta) - \log D_d(\alpha).$$

A positive margin means that the audited procedure supplies more held-out evidence than the estimated requirement; a negative margin is the remaining log-evidence deficit. Because different folds may select different rules, $S_d(\mathcal{A}, \beta)$ evaluates the search procedure rather than one final fitted rule, and therefore does not certify the rule later selected in Gate 2.

These are finite-data measurements. Measured demand substitutes the evaluated population's estimated base error $\widehat{\pi}_d$ into Equation (1), and a fixed convention handles folds with no false positives (Appendix B.5 and Appendix C.4.1); these estimates do not place an upper bound on the population LR^+.

Gate 1 accounts for uncertainty with a task-cluster bootstrap that repeats selection and gives one-sided nominal bounds ℓ and u on M_d. If $u < 0$, the status is INFEASIBLE-FOR-AUDITED-METHOD—*audited-infeasible* in prose—and VFA recommends stopping or revising the audited path before committing independent calibration data. If $\ell > 0$ and held-out TPR is at least β, the status is FEASIBLE and the procedure proceeds to rule-specific calibration. Otherwise the result is UNCERTAIN, indicating that more evaluation data are needed.

These statuses are finite-data screening outcomes. A positive Gate 1 result does not imply that the rule later selected on FIT will pass Gate 2, while a negative result means only that the audited search procedure has not demonstrated sufficient evidence on the available pilot data; it does not rule out every candidate in the family. Appendix B.5 gives the full inference procedure and limitations, and Section 4.3 uses the separate fold-level conditional analysis described there.

Algorithm 1 Verification Feasibility Audit

Input: labelled data from population d, reserved in advance into task-grouped pilot P and disjoint C; selection procedure $\mathcal{A}$; targets (α, β); risk-control procedure $\mathcal{R}$

Gate 1: Held-out evidence-feasibility screen on P (Section 3.1)

task-grouped cross-fitting: rerun $\mathcal{A}$ per fold, aggregate out-of-fold decisions for held-out TPR and FPR
$M_d \leftarrow \log S_d(\mathcal{A}, \beta) - \log D_d(\alpha)$; $(\ell, u) \leftarrow$ selection-aware task-cluster bootstrap bounds
if $u < 0$ **then return** STOP-AUDITED-PATH
else if $\ell \leq 0$ **or** held-out $\mathrm{TPR} < \beta$ **then return** MEASURE-MORE
end if

Gate 2: Rule-specific calibration support on C (Section 3.2)

split C once into task-disjoint FIT/CAL/TEST; fix one FIT-eligible rule (m, τ) on FIT
if no FIT-eligible candidate exists **then return** NO-USEFUL-RULE; no rule is calibrated or tested
end if
if $\mathcal{R}$'s assumptions do not hold for this rule **then**
    $action \leftarrow$ NO-USEFUL-RULE ▷ assumption failure; more calibration data would not change it
else
    apply $\mathcal{R}$ on CAL: obtain $U_{\mathcal{R}}$ and $\widehat{\mathrm{TPR}}_{\mathrm{cal}}$
    if $U_{\mathcal{R}} > \alpha$ **then**
        if $\mathcal{R}$ reports a support limitation **then** $action \leftarrow$ COLLECT-MORE-CALIBRATION
        else $action \leftarrow$ NO-USEFUL-RULE
        end if
    else if $\widehat{\mathrm{TPR}}_{\mathrm{cal}} < \beta$ **then** $action \leftarrow$ NO-USEFUL-RULE
    else if $\mathcal{R}$ supports only a screening statement **then** $action \leftarrow$ CALIBRATION-SCREEN-PASSES
    else $action \leftarrow$ SELECTIVE-USE
    end if
end if
evaluate the fixed rule once on TEST, leaving $action$ unchanged
return $action$ with TEST accepted risk, TPR, and coverage

Output: audit action, with TEST diagnostics whenever Gate 2 evaluates a fixed rule

3.2 GATE 2: CALIBRATION SUPPORT FOR A FIXED RULE

Gate 2 addresses the rule-level question that remains after the evidence-feasibility screen: once one acceptance rule is fixed, what do independent calibration data support about that rule? Gate 1 and Gate 2 therefore evaluate different statistical objects by design. Gate 1 asks whether the au-

dited search family can demonstrate sufficient discrimination relative to the population requirement; Gate 2 asks whether one rule selected from that family can be supported by finite calibration data.

The three task-disjoint roles inside the reserved set C divide this second task. FIT fixes the scoring method, aggregation rule, and threshold, keeping the eligible candidate that accepts the most outputs. A candidate is *FIT-eligible* when it meets both targets on FIT (Appendix B.6 gives the eligibility test and tie rule). The prespecified risk-control procedure $\mathcal{R}$ then evaluates that fixed rule on CAL without retuning it, and TEST is used only for diagnostics.

The selection criteria in the two gates need not coincide. Gate 1 searches for evidence because its purpose is to determine whether the audited family reaches the population feasibility boundary. Once that condition is sufficiently supported, Gate 2 instead favors deployment utility by selecting the highest-coverage FIT-eligible rule and then assessing that fixed rule on independent calibration data. A Gate 1 pass therefore motivates calibration; it does not predict which rule Gate 2 will select or guarantee that the selected rule will be supported.

$\mathcal{R}$ is an input to VFA rather than a calibration method introduced by the framework. It specifies what statistical statement the calibration data support, together with the assumptions and confidence target required for that statement. When $\mathcal{R}$ returns an upper bound $U_{\mathcal{R}}$ on accepted risk, CAL supports the target only if

$$U_{\mathcal{R}} \leq \alpha \qquad \text{and} \qquad \widehat{\mathrm{TPR}}_{\mathrm{cal}} \geq \beta.$$

The retention condition prevents a rule from appearing useful simply by accepting almost nothing.

The strength of the conclusion depends on $\mathcal{R}$. *Empirical accepted risk* is a point estimate, a *large-sample screen* gives an approximate statement, and *finite-sample certification* provides a guarantee under the procedure's assumptions. VFA reports the level of support justified by the chosen $\mathcal{R}$ and available calibration data rather than treating these statements as interchangeable (Appendix E.2).

Failure to support means only that the chosen procedure does not justify the target on these data. If $\mathcal{R}$ reports insufficient calibration support, VFA recommends COLLECT-MORE-CALIBRATION; otherwise it returns NO-USEFUL-RULE. More calibration can address a support limitation, but not an assumption failure (Appendix E).

Finally, TEST reports the fixed rule's accepted risk, TPR, and coverage without changing the rule or calibration action; accepted risk is undefined if nothing is accepted. If no FIT-eligible candidate exists, there is no rule to calibrate or test (Appendix B.6).

The two gates address related but distinct requirements. Gate 1 provides a held-out finite-data screen of whether the search procedure $\mathcal{A}$ supplies enough likelihood-ratio evidence for the population-level risk and retention targets. Gate 2 then asks whether the available calibration data and procedure $\mathcal{R}$ support deploying the rule selected on FIT. Evidence feasibility can therefore hold while finite-sample calibration remains inconclusive.

VFA avoids label reuse by using pilot data P for Gate 1 and a disjoint reserved set C for the FIT/CAL/TEST roles of Gate 2. In a full sequential audit, both gates are interpreted relative to the prespecified deployment population and targets. Component analyses using different scored units or populations remain component-level results rather than an end-to-end certificate. Algorithm 1 summarizes this data flow; NO-USEFUL-RULE refers only to what the chosen $\mathcal{R}$ supports for the audited rule.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks. Our experiments use GSM8K (Cobbe et al., 2021), MATH-500 (Lightman et al., 2024; Hendrycks et al., 2021), BFCL (Patil et al., 2025), IFEval (Zhou et al., 2023), TravelPlanner (Xie et al., 2024), HotpotQA (Yang et al., 2018), Spider (Yu et al., 2018), SWE-bench and SWE-bench-Verified (Jimenez et al., 2024), RAGTruth (Niu et al., 2024), and τ-bench (Yao et al., 2025). We additionally construct Complexity from CodeComplex (Baik et al., 2025), Regex from KB13 (Kushman & Barzilay, 2013), and Bash from NL2Bash (Lin et al., 2018); Temporal is synthesized from date-arithmetic tasks. AppWorld (Trivedi et al., 2024) is used only in the separate pilot-to-sealed-test case study(Section 4.5). The Gate-1, scaling, calibration, and case-study analyses use

partially overlapping subsets. Table 5 and Appendix B.1 give the study membership, evaluation-set sizes, and scored units.

Verifier models and methods. We use five LLMs as judges: GPT-5.4 and GPT-5.4 mini (OpenAI, 2026a;b), Claude Opus 4.7 (Anthropic, 2026), Gemini 3.1 Pro (Google, 2026), and Grok 4.20 (xAI, 2026). We then evaluate different ways of using their judgments: single-verifier selection, per-domain routing, repeated judging, multi-verifier panels, and equal-weight or learned aggregation. Repeated judging and panels adapt self-consistency (Wang et al., 2023) and multi-judge verification (Verga et al., 2024); we also evaluate compatible paths from LLM-as-a-Verifier (Kwok et al., 2026). Tables 6 and 7 and Appendices B.2 and B.3 gives the model, pool, and method details.

Shared design. Unless stated otherwise, we use an accepted-risk target of $\alpha = 0.10$ and a retention target of $\beta = 0.50$ on TPR; these are evaluation targets, not guarantees that every experiment satisfies them. All data splits and resampling are performed at the parent-task level. We reuse cached verifier decisions across analyses, which broadens the range of judgments we can compare without creating additional independent samples. Most experiments below evaluate individual components of VFA rather than running the full sequential audit in Algorithm 1 on every benchmark (Table 4) and Appendices C–F give the full results, sensitivity analyses, and case-study details.

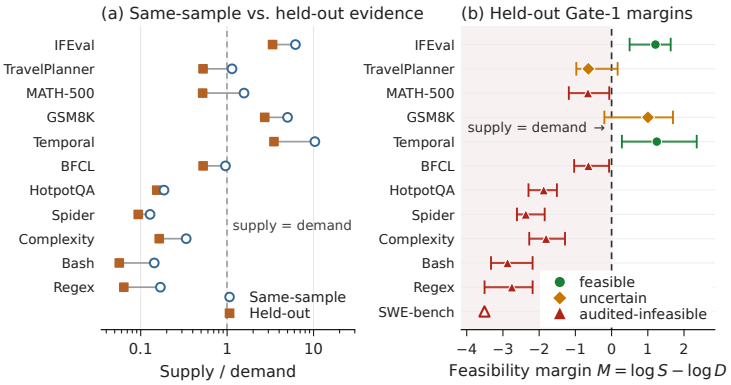


Figure 2: **Evaluation protocol changes apparent verifier evidence. (a)** Same-sample and held-out supply relative to demand on 11 paired sets; the broader comparison is descriptive, while the controlled label-reuse analysis is in Appendix C.2. **(b)** Held-out Gate-1 margins for all 12 sets with selection-aware bounds; zero marks supply equal to demand. SWE-bench is shown without an interval because its bootstrap band is not dependable (Appendix C.1).

Table 1: **Balanced-accuracy gains often leave audit status unchanged.** Each study uses its own matched single-verifier baseline. Δ denotes the change in BalAcc (percentage points), and the last column counts benchmarks with higher BalAcc but unchanged audit status.

Method	BalAcc	Δ (pp)	Status unchanged
Single-verifier baseline	0.7416	—	—
PAR-D routing	**0.7981**	**+5.65**	8/12
Naive-Bayes aggregation	0.7894	+4.78	7/12
L_2-shrunk logistic	0.7892	+4.76	8/12
Supervised weighted vote	0.7852	+4.36	7/12
Three-voter panel	0.7643	+2.27	7/12
Equal-weight vote fraction	0.7638	+2.22	5/12
Multi-style panel	0.7634	+2.18	7/12
Five-sample self-consistency	0.7063	-3.53	6/12
Task-level studies with separate matched baselines			
Cross-vendor panel	0.7735	+0.10	3/8
Rationale-feature score	0.6886	+1.18	2/9

4.2 GATE 1: THE EVALUATION PROTOCOL CHANGES APPARENT EVIDENCE

As discussed in Section 2, verifier selection is not always fully held out from the reported evaluation data. Figure 2 shows why this matters: panel (a) compares same-sample and held-out evidence, while panel (b) shows the resulting held-out Gate-1 margins.

We first isolate label reuse alone. Holding the candidate procedure and evaluated records fixed, we vary only whether evaluation labels are available during selection. Reuse increases measured evidence by a median factor of $1.13\times$ and moves the point margin from negative to positive in 3 of 40 conditions (Appendix C.2); these are point-margin crossings, not uncertainty-aware status changes. The broader comparison in Figure 2a also changes the candidate search and is therefore descriptive, but held-out supply is lower on all 11 paired sets by a median factor of $1.9\times$. On MATH-500, same-sample supply exceeds demand (6.52 vs. 4.15), whereas held-out supply on the same 95 outputs falls to 2.16, giving $M = -0.65$ with an upper bound below zero.

Under held-out evaluation, most populations still lack enough evidence (Figure 2b). IFEval and Temporal are FEASIBLE, GSM8K and TravelPlanner are UNCERTAIN, and seven other settings have nominal upper margin bounds below zero. SWE-bench also has a negative point margin, but its bootstrap classification is not dependable (Table 9 and Appendix B.5). These outcomes are specific to the evaluated populations; Appendix C.3 examines their sensitivity to base error. Synthetic experiments with known procedure-level margins further evaluate the operating behavior of the selection-aware Gate-1 inference procedure (Appendix C.5).

4.3 GATE 1: VERIFIER IMPROVEMENTS AND THE REMAINING EVIDENCE GAP

Better verifier metrics often do not change the benchmark classification. Across 117 matched benchmark–method comparisons, 60 improve balanced accuracy while leaving the classification unchanged. Our per-domain selector PAR-D, for example, chooses the best verifier for each domain using FIT data rather than using one verifier across all domains. It improves macro balanced accuracy by 5.65 percentage points, yet preserves the baseline classification on 8 of 12 benchmarks (Table 1). Each method is compared with the single-verifier baseline from the same pool on the same eligible records. [1]

None of the 36 comparisons whose baseline is classified as *audited-infeasible* in this scaling analysis is classified as feasible. Some methods do resolve uncertainty: voting and learned aggregation move the uncertain Complexity baseline to feasible, while rationale-feature scoring moves TravelPlanner from *audited-infeasible* to UNCERTAIN (Appendix D). The cross-vendor panel shows a similar pattern: macro balanced accuracy improves by 0.10 percentage points, but the benchmark classification is unchanged on all eight benchmarks (Appendix D.2).

The improvements that do increase the margin usually close only a small part of the deficit. Of the 36 baselines classified as *audited-infeasible* in this scaling analysis, 24 show no margin improvement and 12 show a positive gain. For those 12, the median gap closure $\Delta M/(-M_0)$ is 12.5%. The median starting deficit is 1.97 log units, meaning the likelihood ratio would need to increase by roughly sevenfold to reach the boundary (Table 13). Numerical sensitivity checks are reported in Appendix D.4.

4.4 GATE 2: WHAT CALIBRATION DATA SUPPORT

Gate 2 asks what risk statement the available calibration data support for a fixed rule. We evaluate eight rules fixed on FIT under the prespecified protocol of Appendix E. On CAL, all eight satisfy the retention requirement $\beta = 0.50$, and seven have empirical accepted risk at or below $\alpha = 0.10$. **After accounting for calibration uncertainty, only five of the eight pass the prespecified large-sample screen.** Complexity and MATH-500 meet the empirical risk target but fail the screen, while Regex fails both (Table 2).

[1] Section 4.2 audits the full selection procedure using pooled out-of-fold decisions and selection-aware inference. Here we evaluate fold-specific fixed rules and combine the five fold-level results into one benchmark classification for the scaling study. These classifications summarize the fold-specific analysis and are not population-level Gate-1 confidence statements (Appendix B.5).

Stronger finite-sample support is not established at the available sample sizes. None of the three finite-sample analyses supports an applicable rule (Appendix E.2). This largely reflects limited statistical resolution: for example, 100 of the 115 evaluated conditions fall below the necessary task-count threshold for the Hoeffding analysis. These failures do not show that the rules violate the risk target; they show that the tested procedures and available data do not support the stronger claim.

The five large-sample passes should therefore be read as screen passes, not finite-sample guarantees, because the approximation has sub-nominal simulated coverage (Appendix E.3). TEST is diagnostic only: Complexity has no accepted errors on TEST, but this does not change its CAL decision.

Table 2: **Calibration uncertainty changes two of eight decisions.** Observed risk is empirical accepted risk on CAL; Upper bound is its one-sided task-clustered upper confidence bound. "Emp." requires observed risk ≤ 0.10; "Screen" additionally requires the upper bound ≤ 0.10 and CAL TPR ≥ 0.50. All eight rules meet the TPR requirement; TEST risk is diagnostic only.

Benchmark	CAL risk	CAL upper bound	Emp.	Screen	TEST risk
GSM8K	0.023	0.061	Pass	Pass	0.053
Temporal	0.000	0.083	Pass	Pass	0.000
IFEval	0.063	0.086	Pass	Pass	0.061
BFCL	0.056	0.090	Pass	Pass	0.037
Bash	0.055	0.096	Pass	Pass	0.098
Complexity	0.074	0.134	Pass	No	0.000
MATH-500	0.083	0.181	Pass	No	0.059
Regex	0.120	0.211	No	No	0.087

4.5 AUDIT CASE STUDIES

IFEval. On a task-disjoint IFEval split, the primary target $\alpha = 0.10$ leaves Gate 1 uncertain, so VFA returns MEASURE-MORE without entering Gate 2. As a retrospective sensitivity analysis, we relax the target to $\alpha = 0.20$; Gate 1 then becomes FEASIBLE, allowing the audit to proceed through the full FIT/CAL/TEST pipeline. The rule fixed on FIT passes the large-sample calibration screen on CAL, but the finite-sample procedures remain inconclusive, so VFA returns COLLECT-MORE-CALIBRATION rather than selective use. On TEST, the frozen rule has realized accepted risk 0.091. We show more details in Appendix F.1. This example exercises the full two-gate VFA data flow and illustrates that passing Gate 1 does not imply sufficient finite-sample calibration support at Gate 2. The relaxed $\alpha = 0.20$ setting is a target-sensitivity walkthrough rather than the primary evidence-feasibility result.

AppWorld. A separate AppWorld case tests Gate 1 prospectively rather than exercising the full two-gate pipeline. We freeze the verifier panel and its stop decision on a 50-task pilot before opening a 118-task sealed test set. Although measured panel supply rises from 0 on the pilot to 12.875 on the sealed split, it remains below the corresponding demand, and the frozen margin-based diagnostic returns the same stop action (Appendix F.2). Thus, AppWorld provides a prospective check of Gate 1 decision stability, complementary to the retrospective end-to-end IFEval walkthrough.

5 DISCUSSION AND CONCLUSION

Conventional metric progress and decision-level evidence are distinct quantities. Gate 1 separates measurement uncertainty from the measured evidence deficit we call the verification squeeze, while Gate 2 separates favorable empirical accepted risk from a statistically supported statement about one fixed rule. Verifier evaluation should therefore report the evaluated population, the selection protocol, the measured evidence margin, and the supported risk statement alongside its accuracy numbers.

These conclusions are population-conditional and scoped to the tested verifier families, evidence channels, and prespecified fold-level analysis. Richer evidence and sharper finite-sample risk control may change the audit outcome.

REPRODUCIBILITY STATEMENT

Appendix B.1 gives the evaluation populations, scored-output counts, and study membership; Appendix B.2 the model identifiers and inference settings; and Appendix B.3 and Appendix B.3.3 the verifier prompts, score definitions, and pseudocode for every evaluated mechanism. Appendix B.4 gives fold and threshold-candidate construction, Appendices B.5 and B.6 the Gate-1 and Gate-2 estimation conventions, and Appendices C and E the sensitivity analyses that qualify the reported results. Every reported benchmark result comes from a frozen cached artifact and is specific to the model versions and inference settings of Appendix B.2; Appendix B.1 records the evaluation sets whose generation, checker, or source-to-pool provenance is incomplete, which limits re-derivation for those sets.

AI USE STATEMENT

During the preparation of this work, the authors used generative AI tools exclusively for language editing, grammar correction, and polishing the manuscript to improve readability. The tools were not used to produce scientific content. The authors reviewed and edited the final text and take full responsibility for the integrity, claims, and conclusions of this paper.

REFERENCES

Mayur Akewar and Ravi Ranjan. SafeCommit: Certifying When Memory-Grounded Agents May Safely Act, 2026. URL https://arxiv.org/abs/2608.04289. arXiv:2608.04289.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal Risk Control. In *International Conference on Learning Representations*, 2024. arXiv: 2208.02814.

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. doi: 10.1214/24-AOAS1998. arXiv: 2110.01052.

Anthropic. Introducing Claude Opus 4.7. https://www.anthropic.com/news/claude-opus-4-7, April 2026. Model release announcement dated 2026-04-16. Accessed 2026-09-16.

Adrian Arnaiz-Rodriguez, Miguel Baidal, Erik Derner, Jenn Layton Annable, Mark Ball, Mark Ince, Elvira Perez Vallejos, and Nuria Oliver. Between help and harm: An evaluation study of mental health crisis handling by large language models. *JMIR Mental Health*, 13:e88435, 2026. doi: 10.2196/88435. arXiv:2509.24857.

Sher Badshah, Ali Emami, and Hassan Sajjad. SCOPE: Selective Conformal Optimized Pairwise LLM Judging. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. ICML 2026 poster; arXiv:2602.13110.

Seung-Yeop Baik, Joonghyuk Hahn, Jungin Kim, Aditi, Mingi Jeon, Yo-Sub Han, and Sang-Ki Ko. CodeComplex: Dataset for worst-case time complexity prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 19616–19638, Suzhou, China, 2025. Association for Computational Linguistics.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-Free, Risk-Controlling Prediction Sets. *Journal of the ACM*, 68(6):43:1–43:34, 2021. doi: 10.1145/3478535. arXiv: 2101.02703.

Stephen Bates, Emmanuel Candès, Lihua Lei, Yaniv Romano, and Matteo Sesia. Testing for Outliers with Conformal p-values. *Annals of Statistics*, 51(1):149–178, 2023. doi: 10.1214/22-AOS2244.

Bei Chen, Fengji Zhang, Anh Nguyen, Daoguang Zan, Zeqi Lin, Jian-Guang Lou, and Weizhu Chen. CodeT: Code Generation with Generated Tests. In *International Conference on Learning Representations*, 2023. ICLR 2023; arXiv:2207.10397.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, 2021. URL https://arxiv.org/abs/2110.14168. arXiv: 2110.14168.

Jesse Davis and Mark Goadrich. The Relationship Between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pp. 233–240, 2006. doi: 10.1145/1143844.1143874.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 28(1):20–28, 1979. doi: 10.2307/2346806.

Yu Fan, Jingwei Ni, Jakob Merane, Yang Tian, Yoan Hermstrüwer, Yinya Huang, Mubashara Akhtar, Etienne Salimbeni, Florian Geering, Oliver Dreyer, et al. Lexam: Benchmarking legal reasoning on 340 law exams. In *International Conference on Learning Representations*, volume 2026, pp. 43451–43489, 2026.

Yonatan Geifman and Ran El-Yaniv. Selective Classification for Deep Neural Networks. In *Advances in Neural Information Processing Systems 30*, pp. 4885–4894, 2017. arXiv: 1705.08500.

Google. Gemini 3.1 Pro preview. https://ai.google.dev/gemini-api/docs/models/gemini-3.1-pro-preview, 2026. Gemini API model documentation. Accessed 2026-09-16.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. In *International Conference on Learning Representations*, 2024. ICLR 2024; arXiv:2305.11738.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. arXiv: 2103.03874.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024.

Ying Jin and Emmanuel J. Candès. Selection by Prediction with Conformal p-values. *Journal of Machine Learning Research*, 24(244):1–41, 2023.

Jaehun Jung, Faeze Brahman, and Yejin Choi. Trust or Escalate: LLM Judges with Provable Guarantees for Human Agreement. In *International Conference on Learning Representations*, 2025. ICLR 2025.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing Fine-grained Evaluation Capability in Language Models. In *International Conference on Learning Representations*, 2024a. arXiv:2310.08491 [cs.CL].

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4334–4353. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.emnlp-main.248. arXiv: 2405.01535.

Nate Kushman and Regina Barzilay. Using semantic unification to generate regular expressions from natural language. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 826–836, Atlanta, Georgia, 2013. Association for Computational Linguistics.

Jacky Kwok, Shulu Li, Pranav Atreya, Yuejiang Liu, Yixing Jiang, Chelsea Finn, Marco Pavone, Ion Stoica, and Azalia Mirhoseini. LLM-as-a-Verifier: A General-Purpose Verification Framework, 2026. URL https://arxiv.org/abs/2607.05391. arXiv:2607.05391.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. Generative Judge for Evaluating Alignment. In *International Conference on Learning Representations*, 2024. arXiv: 2310.05470.

Shalev Lifshitz, Sheila A. McIlraith, and Yilun Du. Multi-Agent Verification: Scaling Test-Time Compute with Multiple Verifiers. In *Conference on Language Modeling*, 2025. arXiv:2502.20379.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Xi Victoria Lin, Chenglong Wang, Luke Zettlemoyer, and Michael D. Ernst. NL2Bash: A corpus and semantic parser for natural language interface to the Linux operating system. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018. European Language Resources Association.

Xuannan Liu, Zekun Li, Zheqi He, Peipei Li, Shuhan Xia, Xing Cui, Huaibo Huang, Xi Yang, and Ran He. Video-SafetyBench: A Benchmark for Safety Evaluation of Video LVLMs. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2025. doi: 10.52202/085713-4587. arXiv:2505.11842.

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. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.153. arXiv:2303.16634 [cs.CL].

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.

Christopher Mohri and Tatsunori Hashimoto. Language Models with Conformal Factuality Guarantees. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 36029–36047. PMLR, 2024. _eprint: 2402.10978.

Ansong Ni, Srini Iyer, Dragomir Radev, Ves Stoyanov, Wen-tau Yih, Sida I. Wang, and Xi Victoria Lin. LEVER: Learning to Verify Language-to-Code Generation with Execution. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 26106–26128. PMLR, 2023. ICML 2023; arXiv:2302.08468.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10862–10878, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.585.

OpenAI. Introducing GPT-5.4. `https://openai.com/index/introducing-gpt-5-4/`, 2026a. Model release announcement. Accessed 2026-09-16.

OpenAI. Introducing GPT-5.4 mini and nano. `https://openai.com/index/introducing-gpt-5-4-mini-and-nano/`, 2026b. Model release announcement. Accessed 2026-09-16.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The Berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48371–48392. PMLR, 2025.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi Jaakkola, and Regina Barzilay. Conformal Language Modeling. In *International Conference on Learning Representations*, 2024. _eprint: 2306.10193.

Shuvom Sadhuka, Drew Prinster, Clara Fannjiang, Gabriele Scalia, Bonnie Berger, Aviv Regev, and Hanchen Wang. E-valuator: Reliable Agent Verifiers with Sequential Hypothesis Testing, 2025. URL https://arxiv.org/abs/2512.03109. arXiv:2512.03109.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning. In *International Conference on Learning Representations*, 2025. arXiv:2408.03314.

Rakshith S. Srinivasa, Zora Che, Chen Bo Calvin Zhang, Diego Mares, Ernesto Hernandez, Jayeon Park, Dean Lee, Guillermo Mangialardi, Charmaine Ng, Ed-Yeremai Hernandez Cardona, Anisha Gunjal, Yunzhong He, Bing Liu, and Chen Xing. TutorBench: A Benchmark To Assess Tutoring Capabilities Of Large Language Models, 2025. URL https://arxiv.org/abs/2510.02663. arXiv:2510.02663.

Jiayuan Su, Jing Luo, Hongwei Wang, and Lu Cheng. API Is Enough: Conformal Prediction for Large Language Models Without Logit-Access. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 979–995. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.54. _eprint: 2403.01216.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.850.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models, 2024. URL https://arxiv.org/abs/2404.18796. arXiv:2404.18796.

PeiFeng Wang, Austin Xu, Yilun Zhou, Caiming Xiong, and Shafiq Joty. Direct Judgement Preference Optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1979–2009, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.103. arXiv:2409.14664.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large Language Models are not Fair Evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 9440–9450. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.511. arXiv: 2305.17926.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations*, 2023. _eprint: 2203.11171.

Zhilin Wang, Jaehun Jung, Ximing Lu, Shizhe Diao, Ellie Evans, Jiaqi Zeng, Pavlo Molchanov, Yejin Choi, Jan Kautz, and Yi Dong. ProfBench: Multi-Domain Rubrics requiring professional knowledge to answer and judge. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 5355–5384, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35*, 2022. doi: 10.52202/068431-1800.

xAI. Grok 4.20 Non Reasoning. https://docs.x.ai/developers/models/grok-4.20-non-reasoning, 2026. Model documentation. Accessed 2026-09-16.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. TravelPlanner: A benchmark for real-world planning with language agents. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54590–54613. PMLR, 2024.

Jian Xu, Yanning Wu, Delu Zeng, John Paisley, and Qibin Zhao. Look Again Before You Abstain: Budgeted Conformal Evidence Acquisition for Reliable Vision-Language Model, 2026. URL https://arxiv.org/abs/2606.16667. arXiv:2606.16667.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. In *International Conference on Learning Representations*, 2025. arXiv: 2406.12045.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3911–3921, 2018. doi: 10.18653/v1/D18-1425.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating Large Language Models at Evaluating Instruction Following. In *International Conference on Learning Representations*, 2024. arXiv: 2310.07641.

Jitian Zhao, Changho Shin, Tzu-Heng Huang, Satya Sai Srinath Namburi GNVV, and Frederic Sala. CARE: Confounder-Aware Aggregation for Reliable LLM Evaluation. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. Accepted paper.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36*. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-2020. arXiv:2306.05685 [cs.CL].

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-Following Evaluation for Large Language Models, 2023. URL https://arxiv.org/abs/2311.07911. arXiv: 2311.07911.

Yefan Zhou, Austin Xu, Yilun Zhou, Janvijay Singh, Jiang Gui, and Shafiq Joty. Variation in Verification: Understanding Verification Dynamics in Large Language Models. In *International Conference on Learning Representations*, 2026. ICLR 2026; arXiv:2509.17995.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned Large Language Models are Scalable Judges. In *International Conference on Learning Representations*, 2025. arXiv:2310.17631 [cs.CL].

The appendix follows the three empirical questions in the main text. Appendix A defines the notation, and Appendix B gives the populations, verifier methods, data splits, and statistical procedures. Appendix C examines selection and held-out evidence; Appendix D measures the evidence gained from verifier improvements; and Appendix E compares the support provided by different calibration procedures. Appendix F reports the IFEval and AppWorld case studies. Appendix G documents the related-work selection protocols discussed in Section 2.

A NOTATION

Table 3 collects the notation used in the main text and appendix. A hat denotes a plug-in estimate. The supply S_d and margin M_d are the measured audit summaries defined in Section 3.1, not population parameters. We distinguish the Gate-1 selection fold, denoted SEL below, from the independent Gate-2 calibration data, denoted CAL.

Table 3: Notation for the two-gate audit.

Symbol	Meaning
x, Y, z_i	Candidate record and offline correctness label: $Y = 1$ is correct; z_i is the label of record i.
d, π_d, $\widehat{\pi}_d$	Evaluated population, its base error $\Pr(Y = 0 \mid d)$, and the estimated base error.
m, s_m, τ	Verifier method, score, and acceptance threshold.
$A_{m,\tau}(x)$	Acceptance indicator $\mathbf{1}\{s_m(x) \geq \tau\}$; 1 means PASS.
TPR, FPR	Acceptance rates conditional on a correct output and a wrong output, respectively.
R_{acc}	Accepted risk $\Pr(Y = 0 \mid A_{m,\tau} = 1)$; undefined if nothing is accepted.
Coverage	Acceptance probability $\Pr(A_{m,\tau} = 1)$, not confidence-bound coverage.
α, β	Accepted-risk target and TPR retention target; primarily 0.10 and 0.50.
$\mathcal{A}$	Complete method-and-threshold selection procedure; returns a rule (m, τ).
$\mathcal{D}_{\text{sel}}$, $\mathcal{D}_{\text{eval}}$	Disjoint selection and evaluation samples.
$\mathcal{P}$, $\mathcal{C}$	Task-disjoint pilot and reserved data in a full sequential audit.
FIT/SEL/TEST	Gate 1: construct candidates, select a rule, evaluate on the outer held-out fold.
FIT/CAL/TEST	Gate 2: fix a rule, assess calibration support, report diagnostics.
LR^+, $D_d(\alpha)$	Likelihood ratio TPR/FPR and the evidence required by the risk target.
$S_d(\mathcal{A}, \beta)$	Measured evidence from pooled out-of-fold acceptance decisions.
$M_d(\mathcal{A}, \alpha, \beta)$	Measured feasibility margin $\log S_d - \log D_d$.
ℓ, u	One-sided nominal 95% margin bounds; together, a central 90% band.
$\mathcal{R}$, $U_{\mathcal{R}}$	Risk-control procedure and, when supplied, its accepted-risk upper bound.
G, G_{pos}	Calibration task clusters, and those with at least one accepted output.
m_t, A_t, E_t	Scored outputs, accepted outputs, and accepted errors for task t.
$\eta = \delta$	Per-rule one-sided confidence parameter, set to 0.05.

B EXPERIMENTAL SETUP AND ANALYSIS PROTOCOLS

This appendix explains what each experiment evaluates and which data it uses. Table 4 summarizes the studies. The setup proceeds from populations and verifier methods (Appendix B.1, Appendix B.2, and Appendix B.3) to selection, inference, and calibration (Appendix B.4, Appendix B.5, and Appendix B.6). Appendix B.7 describes the separate AppWorld protocol, and Appendix B.8 compares accepted risk with FPR.

The distinction between a full audit and a component study is important. A full sequential VFA reserves task-disjoint pilot data $\mathcal{P}$ and data $\mathcal{C}$ for Gate 2 before evaluating either gate. Most experiments in Section 4 instead evaluate one component on its available task population. Their Gate-1 and Gate-2 results cannot be combined into a sequential guarantee. The IFEval walkthrough explicitly reserves all four roles; AppWorld uses a different pilot-to-sealed-test design.

Table 4: Experimental design and scope. A condition is an evaluated configuration, not an independent sample. The component studies do not constitute sequential two-gate audits.

Study	Design and count	Question addressed
Gate-1 selection	12 evaluation sets; 11 paired same-sample and held-out estimates	Does the evaluation protocol change apparent evidence?
Controlled label reuse	Fixed candidates and evaluated records; 8 benchmarks $\times$ 5 rotations = 40 conditions	How much does access to evaluation labels change the measured margin?
Offline method comparisons	96 main GPT + 8 cross-vendor + 9 rationale-feature = 113 matched comparisons	Do metric gains change the fold-based audit outcome?
Released package	2 settings $\times$ 4 benchmarks = 8 settings; 4 matched	Do the supported package settings close the evidence deficit?
Gate-2 calibration	8 benchmarks $\times$ 5 rotations $\times$ 3 scores = 120 configurations; 115 fixed rules	What do calibration data support for an already fixed rule?
Audit cases	IFEval: 534 tasks reserved four ways. AppWorld: 50 pilot + 118 sealed tasks	What happens when the audit stops, or a panel retains too few correct outputs?

Scale of the cached judgments. The three verifier pools contain 111,267 cached entries: 88,368 in the main GPT pool, 13,203 in the cross-vendor pool, and 9,696 in the rationale-feature pool. The main GPT pool contains 6,312 atoms from 3,958 parent tasks, each evaluated by 14 verifier variants; 88,249 entries have a parsed verdict. The other pools contain 1,467 task-level records evaluated by nine voters and 1,616 task-level records evaluated by six voters, respectively. Five model versions from four providers produced these entries (Table 6). Reusing an entry across aggregators or rotations does not create another independent observation.

B.1 Benchmarks and evaluation sets

Table 5 summarizes the evaluation sets, their study membership, and the numbers of scored outputs and parent tasks. Parent tasks are the units of splitting and resampling; all scored outputs from the same parent task remain in the same fold. A scored output receives a correctness label and an acceptance decision, and we use the term *atom* when one parent task contributes multiple such outputs. Correctness labels are fixed before verifier selection. AppWorld additionally groups tasks into scenarios for resampling (Appendix B.7).

Table 5: Evaluation sets and study membership. G1: held-out selection; M: method comparisons; G2: calibration; Case: AppWorld pilot and sealed evaluation. Counts report loaded scored outputs and parent tasks for each evaluation set. Table 9 separately reports the task-level selection pools and valid-score intersections used by Gate 1.

Source dataset $\rightarrow$ evaluation set	Study	Outputs	Tasks
GSM8K (Cobbe et al., 2021)	G1,M,G2	200	100
MATH-500 (Lightman et al., 2024; Hendrycks et al., 2021)	G1,M,G2	200	100
CodeComplex, Python split (Baik et al., 2025) $\rightarrow$ Complexity	G1,M,G2	420	210
KB13 (Kushman & Barzilay, 2013) $\rightarrow$ Regex	G1,M,G2	459	100
BFCL (Patil et al., 2025)	G1,M,G2	400	100
NL2Bash (Lin et al., 2018) $\rightarrow$ Bash	G1,M,G2	523	100
IFEval (Zhou et al., 2023)	G1,M,G2	825	534
Temporal (synthesized here)	G1,M,G2	200	100
TravelPlanner (Xie et al., 2024)	G1,M	651	180
HotpotQA (Yang et al., 2018)	G1,M	150	150
Spider (Yu et al., 2018)	G1	150	150
SWE-bench (Jimenez et al., 2024)	G1	30	30
SWE-bench-Verified (Jimenez et al., 2024)	M	200	200
RAGTruth (Niu et al., 2024)	M	1350	1350
τ-bench (Yao et al., 2025)	M	884	884
AppWorld (Trivedi et al., 2024)	Case	50 / 118	50 / 118

B.2 MODELS

Table 6 gives the API model identifiers and their roles. Single-shot styles in the main GPT pool use temperature 0. The five-sample B_{sc} and F_{sc} blocks use temperature 0.7, and the three-vote B_w block uses temperature 0.5. AppWorld uses temperature 0 and one call per voter. The method-level inputs and disclosed call counts appear in Table 8.

Table 6: API model identifiers and experimental roles. Table 8 gives method-level inputs and disclosed call counts.

Model identifier	Role and study
`gpt-5.4-mini`	Weak primary judge; TravelPlanner generator; LLM-as-a-Verifier
`gpt-5.4`	Strong primary judge; native task-level forward/backward/critique judges; all AppWorld voters
`claude-opus-4-7`	Cross-vendor API judge
`gemini-3.1-pro-preview`	Cross-vendor API judge
`grok-4.20-experimental-beta-0304-non-reasoning`	Cross-vendor API judge

B.3 VERIFICATION METHODS AND BASELINES

We separate the available judge outputs from the procedures that combine or select them. A *pool* fixes the judge variants and eligible records. A *score* maps those judgments to a numerical value. A threshold then turns that score into an acceptance rule. The methods below are our implementations of the cited ideas; the released external package is described separately in Appendix B.3.5.

All inference is reference-free: the judge sees the task, candidate, and permitted domain information, but not a gold answer. Supervised aggregators use offline correctness labels on FIT; those labels never enter a judge's test-time prompt.

B.3.1 VERIFIER POOLS AND PROMPT ROLES

Table 7 lists the verifier pools and prompt roles. The main GPT pool stores atom-level judgments. The cross-vendor and rationale-feature pools store task-level judgments, with explanations retained in the latter. Each scaling comparison uses a fixed eligible pool and selects its single-voter baseline using that pool's FIT data.

The cross-vendor pool covers Bash, BFCL, Complexity, GSM8K, IFEval, MATH-500, Regex, and Temporal. The rationale-feature pool covers those eight benchmarks and TravelPlanner. External-package scoring covers IFEval, MATH-500, BFCL, and HotpotQA. The main GPT, cross-vendor, and rationale-feature pools are stored as `PRIMARY`, `CROSS-VENDOR`, and `CORE-RATIONALE`, respectively.

B.3.2 SCORE FAMILIES

For record i and voter j, let $z_i \in \{0, 1\}$ be the offline correctness label and let v_{ij} equal 1 for PASS, 0 for FAIL, and 0.5 for a missing verdict. Table 8 summarizes the scores and their inputs. Score fitting and threshold selection are separate operations; their data roles are specified in Appendix B.4.

Supervised weighted voting. Voter j receives weight $w_j = \max\{0, \mathrm{BalAcc}_j^{\mathrm{FIT}} - 0.5\}$, where its FIT balanced accuracy uses only non-missing verdicts. The score is

$$s(i) = \frac{\sum_j w_j v_{ij}}{\sum_j w_j}. \tag{2}$$

If all weights are zero, the score uses equal weights.

Table 7: Verifier pools and prompt roles. Subscripts w and s distinguish the weak and strong model, and sc denotes five-sample self-consistency. Prompts provide the task and candidate but no gold answer. Main GPT prompts also identify the atom being checked. Main GPT forward prompts request a one-word verdict; task-level forward and critique prompts request a JSON verdict.

Pool	Composition
Main GPT	14 verifier variants: $F_w, F_s, P_w, P_s, W_w, W_s, B_w, B_s, F_{sc}, B_{sc}, CoT_s, REF_s, DEV_s, CITE_s$
Cross-vendor	$F_s, CRIT_s, B_s$ for each of Claude, Gemini, and Grok: 3×3 voters
Rationale-feature	Six voters with explanations: $B_s, B_w, CRIT_s, CoT_s, F_s, F_w$
AppWorld	Three single-call voters: $F_s, CRIT_s, B_s$, all using `gpt-5.4`
Style	**Verification role**
Forward (F_w, F_s)	Judge directly whether the candidate is correct for the task.
Backward (B_w, B_s)	Main GPT pool: a yes/no satisfaction probe. Task-level panels: reconstruct and compare — on GSM8K, MATH-500, Temporal, and BFCL assume the candidate is correct, reconstruct a task-stated quantity, and compare it with the task's value; elsewhere, including AppWorld, derive the most task-critical output token independently and compare. Both forms are reference-free.
Per-test (P_w, P_s)	Main GPT pool forward judging of one atom alone, with one domain-specific reasoning hint.
Whole-batch (W_w, W_s)	Main GPT pool forward judging of every atom of a task in one numbered prompt, one `[N] PASS/FAIL` line per atom.
Critique ($CRIT_s$)	Task-level panels. Enumerate substantive correctness or completeness weaknesses, excluding style and presentation, before deciding.
Chain-of-thought (CoT_s)	Reason step-by-step before deciding, adapting the intermediate-reasoning idea of Wei et al. (2022) to a supplied candidate.
Reflection (REF_s)	Three sequential calls: verdict, self-critique of it, final verdict.
Devil's-advocate (DEV_s)	Argue for PASS, then FAIL, then decide which argument is stronger.
Citation-grounded ($CITE_s$)	Quote a span from the candidate, or state that none exists, before deciding; the quoted span is not programmatically checked against the candidate text.

Naive-Bayes aggregation. On FIT, estimate $\mathrm{tpr}_j = (\mathrm{TP}_j + 1)/(n^+ + 2)$ and $\mathrm{fpr}_j = (\mathrm{FP}_j + 1)/(n^- + 2)$. Here n^+ and n^- count correct and wrong records, while TP_j and FP_j sum v_{ij} over the corresponding classes. Missing votes remain at 0.5, so the class denominators are shared by all voters. The score is

$$s(i) = \sum_j \left[v_{ij} \log \frac{\mathrm{tpr}_j}{\mathrm{fpr}_j} + (1 - v_{ij}) \log \frac{1 - \mathrm{tpr}_j}{1 - \mathrm{fpr}_j} \right]. \tag{3}$$

This score treats voters as conditionally independent given correctness. Unlike the latent-label Dawid–Skene model (Dawid & Skene, 1979), it estimates voter reliability directly from FIT labels without an EM step. The logistic score instead fits correctness from the joint vote vector, without this factorization.

Rationale features. The rationale-feature score adds three model-free features to the vote fraction: the logarithm of one plus the mean rationale length in characters, the mean frequency of a fixed hedging lexicon, and the mean agreement between a rationale's independently parsed verdict and its voter's binary prediction. The lexicon includes terms such as *maybe*, *unsure*, and *ambiguous*. Features are standardized on FIT before fitting the logistic model. With fewer than eight FIT records or only one correctness class, the score falls back to the FIT base rate. The extractor processes cached text offline; it does not call another judge.

Repeated judgments and panel scores. Aggregation occurs at two levels. Within a repeated block, sampled judgments are reduced to one cached verdict: five samples for F_{sc} and B_{sc}, adapting self-consistency (Wang et al., 2023), and three votes for B_w. Across blocks, a panel takes the equal-weight fraction of these cached verdicts. Thus, the repeated-verification baseline uses the single cached F_{sc} verdict rather than five separately stored inputs.

Table 8: Verification mechanisms and their scores. Pool membership is fixed within each matched comparison. Score fitting and voter selection use FIT labels only. Gate-1 thresholds are chosen on SEL; Gate-2 thresholds are fixed on FIT before calibration. Calls exclude candidate generation, and a whole-batch call can cover several atoms. Missing aggregation inputs are encoded as 0.5.

Mechanism	Calls / inputs	Score
Voter selection using FIT *labels*		
Single-best voter	1 voter	Raw verdict of the voter with the highest *macro* FIT balanced accuracy across the pool's evaluation sets; one voter everywhere. Scaling baseline.
PAR-D	1 voter per domain	Raw verdict of the voter with the highest FIT balanced accuracy *in that domain* (Algorithm 3).
Label-free vote aggregation		
Equal-weight vote fraction	pool votes	$s_{\text{vote}}(i) = J^{-1}\sum_{j=1}^{J} v_{ij}$ over the whole pool; accept when $s_{\text{vote}}(i) \geq \tau$. Scaling and calibration.
Three-voter panel	3 voters, 5 calls	s_{vote} over the cached B_w, F_w, F_s verdicts.
Multi-style panel	3 voters, 11 calls	s_{vote} over the cached B_{sc}, F_{sc}, P_s verdicts.
Cross-vendor panel	3×3 voters	One joint s_{vote} over all nine vendor–style voters.
Repeated verification	1 voter, 5 calls	The cached F_{sc} verdict itself: the majority of five samples of one fixed candidate, reduced before caching.
AppWorld majority	3 calls	Strict majority of the forward, critique, and backward PASS decisions.
Supervised vote aggregation, fitted on FIT *votes and labels*		
Supervised weighted vote	pool votes	Reliability-weighted vote fraction, Equation (2).
Naive Bayes	pool votes	Summed per-voter log-odds, Equation (3).
Shrunk logistic	pool votes	Predicted correctness probability of an L_2-penalized logistic regression (inverse regularization strength $C = 0.1$) on the unstandardized $\{0, 0.5, 1\}$ vote vector.
Rationale-feature score	6 voters + rationales	L_2-penalized logistic regression ($C = 0.5$) on the vote fraction plus three model-free rationale features, standardized on FIT.
External		
LLM-as-a-Verifier	1 or 3 calls	The released package's own continuous score or mean of three judgments, with its own parser and fallback value 0.5.

Except for AppWorld's fixed majority rule, the panel scores are thresholded at a selected τ; they do not use a fixed majority cutoff. Equality with the threshold is accepted. Differences in model, prompt, and aggregation accompany the call-count differences, so these comparisons are not a controlled compute-scaling curve.

B.3.3 PSEUDOCODE

Algorithm 2 gives the shared protocol for each scaling comparison. Algorithm 3 specifies the per-domain router. Other mechanisms use the same evaluation protocol with the corresponding score from Table 8. A score is fitted on FIT, its threshold is chosen on SEL, and the resulting rule is evaluated once on TEST in each rotation.

The selection fold in this protocol is not Gate-2 calibration data. It chooses the threshold by likelihood ratio. Gate 2 instead fixes the threshold on FIT and uses CAL only to assess the chosen rule (Appendix B.6).

PAR-D selects one voter per domain and rotation, not one voter per example. It is neither an oracle nor a learned domain classifier. No SEL or TEST label enters the voter choice. The matched baseline differs only in the scope of that choice: it maximizes macro FIT balanced accuracy across the pool's domains and therefore uses one voter everywhere. A selected voter can itself be a repeated block, so selecting one voter does not always mean making one API call.

Algorithm 2 Shared scaling evaluation protocol for one benchmark–method comparison

Input: task-grouped folds of one evaluation set; mechanism m; eligible voter pool; targets (α, β)
Output: one fixed rule per rotation and its TEST decisions

for each rotation of Appendix B.4: 3 FIT folds, 1 SEL, 1 TEST **do**
  fit the score s_m of Table 8 on FIT only
  build at most 20 threshold candidates from FIT scores alone, at evenly spaced rank positions over the distinct values
  on SEL, choose τ maximising the likelihood ratio among candidates with $\widehat{\mathrm{TPR}}_{\mathrm{sel}} \geq \beta$; ties take the lowest τ
  if no candidate meets the floor **then**
    take τ with the highest $\widehat{\mathrm{TPR}}_{\mathrm{sel}}$, lowest on ties
  end if
  score TEST once with the fixed (m, τ); record TPR, FPR, demand, and margin
end for
return the fold-level values summarised in Appendix B.5

Algorithm 3 PAR-D: per-domain voter selection

Input: domains $d \in \mathcal{D}$ of one pool, each with its own task-grouped folds; eligible voter pool $\mathcal{J}$ in a fixed order; cached verdicts v_{ij} and labels z_i; TPR target β
Output: per rotation, one voter and one threshold per domain, both fixed before TEST

for each rotation t (three FIT folds, one SEL, one TEST) **do**
  for each domain d **do**
    $\mathrm{BalAcc}^{\mathrm{FIT}}_{j,d} \leftarrow$ balanced accuracy of each voter $j \in \mathcal{J}$ on d's FIT records, non-missing verdicts only ▷ per domain, never pooled
    $j^\star(d,t) \leftarrow \arg\max_j \mathrm{BalAcc}^{\mathrm{FIT}}_{j,d}$, ties by the fixed pool order; $s_d(i) \leftarrow v_{i\, j^\star(d,t)}$ ▷ raw cached verdict; missing scores 0.5
    choose $\tau_{d,t}$ from d's FIT-derived candidates on SEL by the fixed criterion of Algorithm 2, then score d's TEST fold once
  end for
end for
return the domain $\to$ voter map $j^\star(\cdot, t)$ and the TEST decisions

B.3.4 External methods and integration

We evaluate the released LLM-as-a-Verifier implementation (Kwok et al., 2026) with its original prompts, scoring formula, parser, and fallback value. Each cached candidate is presented as a single-step trajectory for absolute scoring by `gpt-5.4-mini`. The two settings use one continuous-score judgment or the mean of three independent judgments. An unparseable response retains the package's fallback score of 0.5.

The criteria-decomposition and pairwise-selection paths were not evaluated. The supported scoring interface lacks criteria inputs, and the cached sets do not uniformly provide two real candidates per task. CARE (Zhao et al., 2026) was also not evaluated: its reported protocol tunes aggregation hyperparameters on a labelled validation subset, and we did not adapt that protocol to these audit experiments.

B.4 Method and threshold selection

Data roles across the studies. The notation $\mathcal{P}$ and $\mathcal{C}$ refers to a full sequential audit, not to an additional split silently applied to every component experiment. The main Gate-1 and scaling studies use their available task populations for cross-fitting. The Gate-2 component study uses a separate allocation of roles within its task population, without a preceding independently reserved pilot. Their results therefore evaluate the two gates separately.

Study	Allocation	Interpretation
Gate-1 selection and scaling	3 FIT, 1 SEL, 1 TEST fold	Approximately 60/20/20 of task groups per rotation; no separate $\mathcal{P}/\mathcal{C}$ reservation.
Gate-2 component study	2 FIT, 2 CAL, 1 TEST fold	Approximately 40/40/20 of task groups; exact primary counts appear in Appendix B.6.
IFEval walkthrough	$\lvert\mathcal{P}\rvert = 115$, $\lvert\mathcal{C}\rvert = 419$	Reserved data: 205 FIT, 102 CAL, 112 TEST tasks. Gate 2 is not reached at $\alpha = 0.10$; at $\alpha = 0.20$ it is.
AppWorld case	50 pilot, 118 sealed tasks	One frozen panel evaluated on a sealed set; not a full Gate-2 allocation.

Gate-1 folds and rotations. A rotation assigns five task-grouped folds to three roles. For rotation t, fold t is the outer TEST fold, fold $(t + 1) \bmod 5$ is the selection fold, and the remaining three folds are FIT. We call the selection fold SEL throughout this appendix; its stored label is CAL, but it is used for selection rather than risk calibration. Fold membership is a deterministic function of the parent-task identifier. All records of a task remain together, and each record serves as outer test data once across the five rotations. The fractions above describe fold assignments, not exact task or record counts in equally sized partitions.

Threshold candidates and selection. FIT constructs at most 20 candidate thresholds at evenly spaced rank positions among the distinct FIT scores. SEL selects the threshold with the largest likelihood ratio among candidates whose selection-fold TPR is at least β. Ties take the lowest threshold. If no candidate meets the TPR floor, the fixed-mechanism protocol takes the threshold with the highest selection-fold TPR, again taking the lowest threshold on ties. The selected rule is then applied once to the outer TEST fold.

Method search in the selection study. The held-out search also chooses among methods using selection-fold likelihood ratio subject to TPR $\geq \beta$. It excludes reference-aware, ground-truth-replay, and placeholder records and requires at least 20 non-null predictions for a candidate. An additional multi-voter vote-fraction candidate requires at least two broadly overlapping voters and 40 common examples. A method whose fallback threshold misses the selection-fold TPR floor cannot win this search. These eligibility requirements are separate from PAR-D's voter choice on FIT.

The same-sample diagnostic uses a different search: it greedily selects a subset of the gold-answer-free cached voters and accepts only unanimous PASS votes. Selection and evaluation both use the full per-domain pool, with at least 40 examples, $\beta = 0.50$, and the add-one FPR convention of Appendix B.5. Because this diagnostic also changes the candidate search, Appendix C.2 provides a separate controlled comparison that changes only evaluation-label reuse.

Fixed mechanisms and Gate-2 selection. Each scaling row evaluates its named mechanism rather than searching across mechanisms. Its matched baseline selects the voter with the highest macro FIT balanced accuracy across the same pool's evaluation sets. Both then follow Algorithm 2.

Gate 2 uses a different selection criterion. It fixes a threshold on FIT alone, maximizing FIT coverage among candidates with empirical accepted risk at most α and empirical TPR at least β. If no candidate meets both conditions, no rule is passed to CAL or TEST. The calibration protocol and its exact primary allocation are given in Appendix B.6.

B.5 GATE 1 INFERENCE

Gate-1 measurements use the base error and acceptance rates of the same evaluated population. TPR divides accepted correct outputs by all correct outputs; FPR divides accepted wrong outputs by all wrong outputs. The selection study pools out-of-fold acceptance decisions before computing supply and margin. The scaling study instead computes fold-level quantities for each fixed mechanism and summarizes them across folds. These are different analyses and their benchmark labels are not interchangeable.

Finite-data likelihood ratios. The reported measurements replace the empirical false-positive rate with

$$\widehat{\mathrm{FPR}}_{+1} = \frac{\mathrm{FP}+1}{n_{\mathrm{wrong}}+1}, \qquad \widehat{\mathrm{LR}}^{+} = \frac{\widehat{\mathrm{TPR}}}{\widehat{\mathrm{FPR}}_{+1}},$$

where FP and n_{wrong} refer to the evaluated records. This convention is applied when false positives are observed as well as when their count is zero. In the scaling analysis it is applied within each test fold. It never increases the empirical likelihood ratio and bounds the reported ratio by $n_{\mathrm{wrong}}+1$. This is a finite-data convention, not a confidence bound on the population likelihood ratio.

For the scaling summaries, likelihood ratios are additionally floored at 10^{-9} before taking logarithms. With a positive add-one FPR, the floor binds when a fold accepts no correct output. A macro margin containing such a fold partly reflects this numerical floor. Appendix D.4 reports its sensitivity. AppWorld uses its own version of this convention in Appendix B.7.

Selection-aware inference. The selection study uses 2,000 parent-task bootstrap replicates. Each replicate repeats method and threshold selection and recomputes base error, demand, supply, and margin. The 5th and 95th percentiles give the lower and upper one-sided nominal 95% bounds; together they form a central 90% band, not a two-sided 95% interval. The uncertainty statement is nominal rather than a finite-sample guarantee.

IFEval and Temporal have positive lower bounds, with bands $[0.498, 1.632]$ and $[0.284, 2.351]$. GSM8K and TravelPlanner have bands spanning zero, $[-0.199, 1.695]$ and $[-0.974, 0.167]$. Seven other benchmarks have nominal upper margin bounds below zero. SWE-bench also has a negative point margin, but its bootstrap classification is not dependable.

A bootstrap replicate with no valid selection receives the point margin. This does not occur for eleven benchmarks. For SWE-bench it occurs in 410 of 2,000 replicates from only 30 source task clusters. Assigning this mass to the negative point margin narrows the band and is not conservative for the recorded audited-infeasible classification. Its reported lower endpoint is unbounded, and neither endpoint should be treated as a dependable confidence bound. Appendix C.1 retains the recorded label for traceability while qualifying its interpretation.

Conditional fold-level inference for scaling. The scaling study also uses 2,000 parent-task bootstrap replicates, but resamples the TEST records within each fold while holding that fold's selected rule and demand fixed. Selection is not repeated in these replicates. A fold is FEASIBLE when its lower margin bound exceeds zero and its observed TPR meets β; it is INFEASIBLE-FOR-AUDITED-METHOD when its upper bound is below zero; otherwise it is UNCERTAIN.

The benchmark summary uses the recorded fold-vote rule. It is FEASIBLE when at least half the folds are feasible and that count is not exceeded by the audited-infeasible count. It is audited-infeasible only under a strict majority; all other splits are uncertain. This convention favors feasible in a tie. Macro margins and gap-closure effect sizes summarize the fold-level margins. The resulting benchmark labels describe these fold summaries, not selection-aware population-level confidence statements.

Appendix C.4.1 and Appendix C.4.2 report false-positive and target sensitivities. Appendix D.4 checks the log floor and the stability of the fold-vote summary. These checks supplement rather than replace the primary selection-aware analysis.

B.6 GATE 2 CALIBRATION PROTOCOL

Allocation and primary analysis. The Gate-2 component study uses five parent-task folds: two for FIT, two for CAL, and one for TEST. Five rotations and three score families—vote fraction, naive Bayes, and shrunk logistic—give 120 planned configurations. The primary analysis is rotation 4 with vote fraction. The other rotations and scores assess sensitivity; they are not searched to select the primary result.

The exact primary allocation is shown below. Counts are parent tasks, not scored outputs. Unequal counts reflect the task-ID fold assignment, so a two-fold role need not contain exactly 40% of the tasks. This component study has no separately reserved Gate-1 pilot.

Primary Gate-2 allocation: parent tasks (rotation 4).

Benchmark	Total	FIT	CAL	TEST
GSM8K	100	39	43	18
Temporal	100	37	39	24
IFEval	534	227	195	112
BFCL	100	36	51	13
Bash	100	37	37	26
Complexity	210	82	81	47
MATH-500	100	47	32	21
Regex	100	36	44	20

One parent task may contribute several outputs. In the constant-size calibration populations, BFCL has four scored outputs per task, while GSM8K, Temporal, Complexity, and MATH-500 have two. IFEval, Bash, and Regex have variable-size task groups. All outputs of a parent task remain in the same role; task counts must not be read as atom counts.

Fixing and assessing the rule. FIT fixes the score and threshold using the criterion in Appendix B.4. A threshold is eligible only when its empirical FIT accepted risk is at most α and its empirical FIT TPR is at least β. Among eligible thresholds, the rule with the greatest FIT coverage is retained. Five configurations, all on Regex, have no eligible threshold; no rule is calibrated or tested in those configurations.

For each of the remaining 115 fixed rules, CAL assesses the chosen risk statement without changing the rule. The action is fixed before TEST, which reports accepted risk, TPR, and coverage even when the rule fails the calibration screen. No TEST outcome enters selection or calibration.

Cluster-level quantities. For calibration task t, let m_t be its number of scored outputs, A_t its number of accepted outputs, and E_t its number of accepted errors. With G task clusters, the pooled accepted-risk estimate is

$$\widehat{R} = \frac{\sum_t E_t}{\sum_t A_t}.$$

This is the error fraction among all accepted outputs, not the average of task-specific error fractions. TPR and coverage use the denominators defined in Section 2. Accepted risk is undefined if nothing is accepted.

Primary large-sample screen. The primary screen uses a cluster-robust, linearized upper confidence bound. Let $u_t = E_t - \widehat{R}A_t$ and $\overline{A} = G^{-1}\sum_t A_t$. Its standard error is

$$\mathrm{SE}(\widehat{R}) = \frac{1}{\overline{A}}\sqrt{\frac{\sum_t u_t^2}{G(G-1)}}.$$

When $G \geq 2$ and $\sum_t u_t^2 > 10^{-9}$, the upper bound is $U = \widehat{R} + t_{G-1,1-\eta}\,\mathrm{SE}(\widehat{R})$, where $t_{G-1,1-\eta}$ is the one-sided Student-t critical value. In the degenerate case with accepted outputs, the reported bound is

$$U = \min\left\{1, \widehat{R} - \frac{\log \eta}{G_{\mathrm{pos}}}\right\},$$

where G_{pos} counts task clusters containing at least one accepted output and $\eta = 0.05$. The screen passes when $U \leq \alpha$ and the observed calibration TPR is at least β. Retention and coverage are empirical utility measurements in this procedure.

Finite-sample comparison. The Hoeffding analysis tests the same accepted-risk target using

$$Z_t = E_t - \alpha A_t \in [-\alpha m_t, (1-\alpha)m_t].$$

For independent task clusters with task sizes fixed independently of response outcomes, the reported one-sided radius is

$$r = \sqrt{\frac{\log(1/\eta)\sum_t m_t^2}{2G^2}}.$$

The target is supported when at least one output is accepted and $\overline{Z} + r \leq 0$. Appendix E.2 gives the task-count diagnostic and the separate Hoeffding–Bentkus and empirical-Bernstein analyses. Their applicability restrictions differ; they must not be assigned a common evaluation denominator.

Scope of the statements. The procedures assume independent task clusters. The primary screen additionally relies on a large-sample approximation, whose simulated coverage is examined in Appendix E.3. Each fixed-rule assessment uses $\eta = \delta = 0.05$ without dividing it by the number of FIT candidates, because the threshold is fixed before calibration. No simultaneous guarantee is claimed across rotations, score families, or benchmarks.

Bash and Regex have reference-determined but variable numbers of atoms, with no task-size bound fixed by the benchmark design. The reported finite-sample implementations do not use the observed sample maximum as a substitute for that bound. Appendix E.2 distinguishes these applicability restrictions from an evaluated rule that fails to receive support.

B.7 AppWorld prospective protocol

AppWorld uses a pilot-to-sealed-test protocol, not a complete Gate-2 FIT/CAL/TEST assessment. Within the benchmark's default split, we selected the configuration with the greatest joint availability of a non-empty cached trajectory and a checker label. This choice did not use success rates, verifier scores, or margins. The 50-task pilot, 118-task sealed set, pilot diagnosis, and recommended action were fixed before the sealed evaluation.

The frozen panel uses forward, critique, and backward judgments with the task-level prompt roles in Appendix B.3.1 and the model settings in Appendix B.2. The backward voter uses output-mask reconstruction. Each voter sees the task instruction and the agent's outgoing API-call trace, but not the application responses, so the panel has no execution evidence. The panel accepts by strict majority and withholds its decision if any vote is missing. Individual voters are reported only as diagnostics.

The frozen diagnostic classifies the margin as feasible when its one-sided 95% lower bound is positive, audited-infeasible when its upper bound is negative, and borderline otherwise. It omits the TPR condition in Algorithm 1. The comparison with the paper's $\beta = 0.50$ target is therefore post hoc for AppWorld.

Likelihood ratios use the add-one FPR convention and are floored at 10^{-9} before taking logarithms, both in point estimates and bootstrap replicates. Resampling uses scenarios: 36 clusters for the 50 pilot tasks and 56 for the 118 sealed tasks. Demand is fixed at its point estimate, and degenerate replicates receive the point margin. Appendix F.2 explains why the resulting zero-false-positive margin remains a diagnostic rather than a population-level infeasibility finding.

B.8 Accepted risk and false-positive rate

FPR and accepted risk condition on different sets of outputs. FPR asks how often a wrong output is accepted; accepted risk asks how often an accepted output is wrong. A small FPR therefore does not by itself establish a small accepted risk.

The cached diagnostic illustrates this distinction using class-conditional conformal p-values from correct and incorrect calibration outputs. At $\alpha = 0.10$, a Bonferroni-corrected one-sided binomial screen finds no significant breach of the wrong-conditioned target for vote fraction. Nevertheless, its accepted sets have pooled accepted risk 0.055 over 527 accepts, with benchmark-specific accepted risks of 0.500 on Regex, 0.375 on BFCL, and 0.268 on Complexity. Failure to detect a breach of an FPR target is not support for an accepted-risk target.

Across three score families, the two observed rates differ in 21 of 24 benchmark–score conditions. The three equalities are on Temporal, where both rates are zero. The largest contrasts occur for two supervised scores on Bash and shrunk logistic on Regex: wrong-conditioned rates of 0.0429, 0.0429, and 0.1316 accompany accepted risks of 0.7500, 1.0000, and 1.0000. These are different summaries of the same acceptance decisions, not contradictory measurements.

C SELECTION AND HELD-OUT EVIDENCE

This appendix supports Section 4.2. Appendix C.1 reports the held-out margins and the records behind them. Appendix C.2 isolates evaluation-label reuse while holding the candidates and evaluated records fixed. Appendix C.3 varies base error at fixed class-conditional rates, and Appendix C.4 checks the false-positive convention and the target choices.

C.1 HELD-OUT RESULTS AND EVALUATION-SET COVERAGE

Table 9 reports the complete held-out results. Its loaded-pool count describes the available task-level records; its audited count describes the valid-score intersection on which the reported margin is measured. The cluster count identifies the parent-task groups used for resampling. A dagger flags an intersection below 90% of the loaded pool or fewer than 40 audited records, as defined in Appendix B.1. This coverage flag does not itself change an audit status.

Table 9: Held-out evidence at $\alpha = 0.10$ and $\beta = 0.50$. Pool n counts loaded task-level records; n counts the audited valid-score intersection; Clust. counts resampled parent-task groups. S^{in} and S^{out} are same-sample and held-out supply, D is demand, and $M = \log S^{\text{out}} - \log D$. The point boundary $\pi_{\max}$ is defined in Appendix C.3. Status uses selection-aware bounds, except that SWE-bench's recorded label is retained only with the qualification in Appendix C.1.

Benchmark	Pool n	n	Clust.	π_d	S^{in}	S^{out}	D	M	$\pi_{\max}$	Out-of-fold status
IFEval	534	505	505	0.145	9.00	5.10	1.52	$+1.21$	0.362	FEASIBLE
TravelPlanner$^{\star\diamond}$	149	145	145	0.172	2.19	0.99	1.88	-0.64	0.099	UNCERTAIN
MATH-500*	100	95	95	0.316	6.52	2.16	4.15	-0.65	0.194	INFEAS.$^{\text{aud}}$
GSM8K$^{\diamond}$	145	142	99	0.324	22.00	11.75	4.31	$+1.00$	0.566	UNCERTAIN
Temporal‡	139	137	99	0.328	42.00	15.33	4.40	$+1.25$	0.630	FEASIBLE
BFCL	139	139	100	0.331	4.55	2.35	4.45	-0.64	0.207	INFEAS.$^{\text{aud}}$
HotpotQA	150	150	150	0.507	1.72	1.42	9.24	-1.87	0.136	INFEAS.$^{\text{aud}}$
Spider	150	150	150	0.573	1.55	1.13	12.09	-2.37	0.112	INFEAS.$^{\text{aud}}$
Complexity†	210	176	176	0.597	4.21	2.18	13.31	-1.81	0.195	INFEAS.$^{\text{aud}}$
Bash†	100	75	75	0.733	3.00	1.40	24.75	-2.87	0.135	INFEAS.$^{\text{aud}}$
Regex	100	95	95	0.758	4.57	1.80	28.17	-2.75	0.167	INFEAS.$^{\text{aud}}$
SWE-bench†§	30	16	16	0.813	—	1.17	39.00	-3.51	0.115	INFEAS.$^{\text{aud}}$

$\star$: same-sample supply exceeds demand but held-out supply does not. $\diamond$: the margin band spans zero. †: reduced valid-score coverage. ‡: same-sample supply reaches the zero-FP measurement limit. §: no valid same-sample estimate and an unreliable bootstrap classification. INFEAS$_{\text{aud}}$ abbreviates INFEASIBLE-FOR-AUDITED-METHOD.

Point margins and uncertainty-aware outcomes. The point margins are positive for GSM8K, IFEval, and Temporal and negative for the other nine populations. Selection-aware inference changes the interpretation of two point readings. GSM8K becomes uncertain despite a positive point margin; TravelPlanner becomes uncertain despite a negative one. IFEval and Temporal remain feasible, and seven populations have nominal upper margin bounds below zero. SWE-bench retains its recorded negative classification only with the bootstrap qualification below. Thus, a positive point margin is not enough to justify proceeding to Gate 2.

Which records are evaluated. The audited intersection retains 53% of the loaded SWE-bench pool and 75–100% of the other pools, with a median of 96%. The median shift in base error from the loaded pool to the audited intersection is 0.003; the largest is 0.054 on SWE-bench, where 14 records are excluded. The nine unflagged sets each exclude at most five records, too few for a useful representativeness comparison.

The broader same-sample comparison does not always use an identical evaluated set. TravelPlanner uses 142 same-sample records with demand 1.92, compared with 145 held-out records with demand 1.88. MATH-500 uses the same 95 records and demand 4.15 in both analyses. Appendix C.2 removes this source of mismatch in its controlled comparison.

SWE-bench qualification. For SWE-bench, selection fails in 410 of the 2,000 bootstrap replicates. Those replicates receive the negative point margin, which narrows the reported band; its lower endpoint is also unbounded. As explained in Appendix B.5, neither endpoint supports a dependable uncertainty-aware classification. Figure 2 therefore plots only its point estimate. The reduced valid-score coverage is a separate limitation, not a remedy for this inference problem.

C.2 Controlled label-reuse comparison

The broader same-sample and held-out comparison changes both the search procedure and, in some cases, the evaluated records. This analysis isolates label reuse. It fixes FIT, the candidate family, the selection objective, and the tie rule. A score- and label-independent task-ID rule matches the numbers of tasks on the selection and evaluation sides.

One rule is selected without access to the evaluation side's labels. The comparison rule uses those labels during selection. Both rules are then measured on the same evaluation records. The only difference being tested is whether selection can use the labels on which its performance will be reported.

Table 10: Controlled effect of evaluation-label reuse. Each benchmark reports its median margin increase over five rotations; the macro row is the median of those eight benchmark medians. The same records are evaluated under both selection protocols. Crossings count negative-to-positive point margins, not uncertainty-aware status changes.

Benchmark	Median ΔM	$\exp(\Delta M)$	Point-margin crossings / rotations
MATH-500	0.251	1.29	1/5
GSM8K	0.194	1.21	1/5
Temporal	0.000	1.00	0/5
BFCL	0.000	1.00	0/5
Complexity	0.138	1.15	0/5
IFEval	0.110	1.12	0/5
Bash	0.647	1.91	0/5
Regex	0.054	1.06	1/5
Macro median (8 benchmarks)	0.124	1.13	3/40 conditions

Across the eight benchmarks, the macro median increase is $\Delta M = 0.124$, equivalent to a $1.13\times$ increase in measured evidence. Three of the 40 benchmark–rotation conditions cross from a negative to a positive point margin: one each on MATH-500, GSM8K, and Regex. None crosses in the opposite direction. These are point-margin crossings, not changes in uncertainty-aware audit status.

This comparison measures the size of label-reuse optimism under a fixed procedure. It does not attribute the full contrast in Table 9 to label reuse, because that broader comparison also changes the search. The direction of the effect partly follows from the selection objective; the empirical findings are its magnitude and crossing frequency.

C.3 Fixed-rate base-error sensitivity

This post-hoc sensitivity analysis holds each audited operating point's measured class-conditional rates fixed and varies only the base error. It asks how the same measured evidence compares with different population-specific requirements, not how the verifier would perform after changing the solver.

At $\alpha = 0.10$, demand and margin are

$$D(\pi) = \frac{9\pi}{1-\pi}, \qquad M(\pi) = \log S - \log D(\pi).$$

For $\pi = 0.05, 0.10, 0.20, 0.30, 0.50, 0.70, 0.90$, demand is respectively $0.47, 1.00, 2.25, 3.86, 9.00, 21.00, 81.00$. This is a $171\times$ range in required evidence without changing the measured verifier rates.

The point-margin boundary is

$$\pi_{\max} = \frac{\alpha S}{1 - \alpha + \alpha S}.$$

It is the largest base error at which the fixed measured supply meets demand. Table 9 reports this value for each population. For the eight rows carrying an audited-infeasible label, including the qualified SWE-bench row, it ranges from 0.112 to 0.207, below each row's evaluated base error. At $\pi = 0.10$, every row except TravelPlanner has a non-negative point margin; TravelPlanner is at -0.01. At $\pi = 0.50$, only GSM8K ($+0.27$) and Temporal ($+0.53$) remain non-negative.

These are point sensitivities, not new confidence statements or revised audit statuses. They assume unchanged class-conditional rates under a class-prior shift; changing the solver need not preserve those rates. The retention requirement also remains separate from the risk boundary.

The scored population must remain consistent within each calculation. For example, BFCL has base error 0.331 in the task-level selection study but 0.500 in the atom-level mechanism study, whose tasks contribute equal numbers of correct and wrong scored outputs. A supply measured in one population cannot be paired with the other population's demand.

C.4 False-positive and target sensitivities

C.4.1 The add-one FPR convention

The add-one convention in Appendix B.5 increases the empirical FPR and decreases the reported likelihood ratio, including when false positives are observed. Two alternative conventions preserve every held-out point-margin sign: an exact one-sided 95% Clopper–Pearson upper bound on FPR and $+0.5$ smoothing. At $\alpha = 0.10$ and $\beta = 0.50$, GSM8K, IFEval, and Temporal retain positive point margins and the other nine remain negative. These point checks do not replace the selection-aware classifications.

No held-out audit operating point has zero observed false positives. The zero-FP limitation instead arises in Temporal's same-sample supply of 42.00 and in the sealed AppWorld analysis (Appendix F.2). The finite-data convention does not bound the corresponding population likelihood ratio.

C.4.2 Conditional target sensitivity

The primary targets, $\alpha = 0.10$ and $\beta = 0.50$, were fixed before measurement. This sensitivity uses conditional inference: within each target setting, the selected operating point and demand are held fixed during resampling rather than reselected in every replicate.

Writing F/I/U for the numbers of feasible, audited-infeasible, and uncertain populations, the counts at $\beta = 0.50$ are $1/9/2$ for $\alpha = 0.05$, $3/9/0$ for $\alpha = 0.10$, and $3/6/3$ for $\alpha = 0.20$. At $\alpha = 0.10$, they are $3/8/1$, $3/9/0$, and $3/9/0$ for $\beta = 0.30, 0.50, 0.70$, respectively. GSM8K, IFEval, and Temporal are feasible at each setting except the strictest risk target, where only IFEval remains feasible.

Under this conditional analysis, the result is more sensitive to the risk target than to the retention target. Its $3/9/0$ count at the primary targets is not the paper's primary Gate-1 classification: Table 9 uses selection-aware inference and records $2/8/2$, with the SWE-bench qualification in Appendix C.1.

C.5 Operating characteristics of Gate-1 inference

Gate 1 uses a selection-aware bootstrap to quantify uncertainty after method and threshold selection. Because these bounds are intended as nominal screening bounds rather than finite-sample certificates, we study their operating behavior in synthetic populations where the corresponding procedure-level margin is known.

We distinguish this target from the oracle family quantity $M_d^\star$. Let $\mathcal{A}(D)$ denote the rule returned by the prespecified search procedure from a fresh sample D with the same size and data-role structure. We define the procedure-level acceptance rates by averaging over both D and a fresh population

draw,

$$\mathrm{TPR}_{\mathrm{proc}} = \Pr_{D,(X,Y)\sim d}[\mathcal{A}(D)(X) = 1 \mid Y = 1], \qquad \mathrm{FPR}_{\mathrm{proc}} = \Pr_{D,(X,Y)\sim d}[\mathcal{A}(D)(X) = 1 \mid Y = 0],$$

and define

$$M_{\mathrm{proc}} = \log \frac{\mathrm{TPR}_{\mathrm{proc}}}{\mathrm{FPR}_{\mathrm{proc}}} - \log D_d(\alpha).$$

This is the population counterpart of the cross-fitted evidence delivered by the search procedure, rather than the performance of an oracle rule chosen from the candidate family.

We evaluate the complete Gate-1 pipeline across 93 synthetic settings, varying the number of task clusters, base error, candidate-family size, selection difficulty, and within-task dependence. Each setting contains 500 independent audit datasets, and each audit uses the same 2,000-replicate selection-aware bootstrap as the empirical study. Method and threshold selection are rerun within the bootstrap, giving 46,500 complete simulated audits in total.

Across the simulated settings, Gate 1 is conservative on the decision to proceed. Among settings with non-positive procedure-level margin, the largest observed frequency of a FEASIBLE decision is 0.8%. Increasing the candidate family from one to 50 methods does not systematically degrade this behavior, nor do near-tied candidate methods or task-level clustering. As the procedure-level margin becomes more positive and the number of task clusters increases, decisions increasingly move from UNCERTAIN toward FEASIBLE, while settings near the feasibility boundary are more often classified as uncertain (Figure 3).

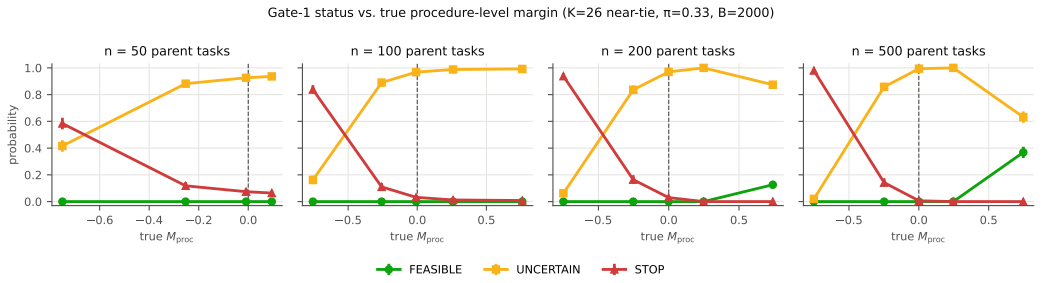


Figure 3: Operating behavior of the Gate-1 selection-aware bootstrap in synthetic populations with known procedure-level margins. Each point summarizes independent applications of the complete Gate-1 procedure, including method and threshold selection. Negative-margin settings predominantly lead to STOP or UNCERTAIN, while positive-margin settings increasingly lead to FEASIBLE as the number of task clusters grows. Uncertainty is concentrated near the feasibility boundary.

The bootstrap bounds remain nominal rather than finite-sample guarantees. Their lower-bound behavior is conservative across the simulated grid, while departures from nominal upper-bound coverage are concentrated in a minority of the smallest-sample and high-base-error settings. We therefore use the bootstrap as an uncertainty-aware screening device and retain the qualified interpretation of Gate-1 statuses from Section 3: a positive result supports proceeding to rule-specific calibration, whereas a negative result indicates that the audited procedure has not demonstrated sufficient evidence on the available data.

D VERIFIER IMPROVEMENTS ACROSS METHODS AND PROVIDERS

This appendix supports Section 4.3 by separating changes in verifier metrics from changes in audit outcome. Appendix D.1 defines the matched comparisons and reports method-level results; Appendix D.2 examines the cross-vendor panel; Appendix D.3 measures improvements relative to the initial evidence deficit; and Appendix D.4 checks the numerical and fold-summary conventions. All results use the fold-level scaling analysis of Appendix B.5, rather than the selection-aware classification in Appendix C.1.

D.1 METHOD COMPARISON

What counts as a comparison. Each comparison pairs a non-baseline method with the single-verifier baseline from the same pool on the same eligible records. The main GPT pool contributes $8 \times 12 = 96$ comparisons, the cross-vendor pool contributes eight, and the rationale-feature pool contributes nine, for 113 offline matched comparisons. Including their 29 baseline entries gives 142 offline benchmark–method entries; that larger number is not a count of comparisons.

The external package adds eight non-baseline settings, of which four have matched baselines. Thus, 121 non-baseline settings were evaluated and 117 support a matched comparison. The four unmatched settings are IFEval and HotpotQA under each of the two package settings. They are excluded because of population or baseline availability, not because of the result they produced.

Metric gains and status changes. Among the 113 offline comparisons, 60 improve balanced accuracy without changing status, 42 preserve status without improving balanced accuracy, and 11 change status. Adding the four matched external settings yields 105 unchanged statuses and 12 changes among 117 comparisons.

The 12 changes comprise three uncertain-to-feasible transitions on Complexity; two uncertain-to-audited-infeasible transitions on Complexity and MATH-500; five feasible-to-uncertain transitions on Bash; one feasible-to-audited-infeasible transition on GSM8K; and one audited-infeasible-to-uncertain transition on TravelPlanner. No audited-infeasible baseline becomes feasible.

Table 11: Method comparisons at $\alpha = 0.10$ and $\beta = 0.50$. Matched baseline macro balanced accuracy is 0.7416 for the main GPT pool, 0.7726 for the cross-vendor pool, and 0.6768 for the rationale-feature pool. F/I/U counts feasible, audited-infeasible, and uncertain fold-summary labels. The final columns count status preservation with a balanced-accuracy gain, preservation without a gain, and any status change.

				Status	Status preserved		Status
Mechanism	Pool	Sets	BalAcc	F/I/U	with gain	no gain	changed
PAR-D (routing)	main GPT	12	**0.7981**	5/3/4	8	3	1
Naive-Bayes aggregation	main GPT	12	0.7894	6/3/3	7	3	2
L_2-shrunk logistic	main GPT	12	0.7892	5/3/4	8	3	1
Supervised weighted vote	main GPT	12	0.7852	6/3/3	7	3	2
Three-voter panel	main GPT	12	0.7643	6/3/3	7	5	0
Equal-weight vote fraction	main GPT	12	0.7638	6/3/3	5	5	2
11-call multi-style panel	main GPT	12	0.7634	6/3/3	7	5	0
5-sample self-consistency	main GPT	12	0.7063	5/5/2	6	4	2
Cross-vendor panel	cross-vendor	8	0.7735	2/4/2	3	5	0
Rationale-feature score	rationale-feature	9	0.6886	0/5/4	2	6	1

Examples behind the aggregate result. On Complexity, the matched baseline is uncertain at balanced accuracy 0.7123. Equal-weight voting, supervised weighted voting, and naive-Bayes aggregation raise balanced accuracy to 0.8063, 0.8068, and 0.8072, with margin gains of 1.171, 1.043, and 1.045, respectively, and move the summary to feasible. Three other methods improve balanced accuracy by 0.0513–0.1304 without changing status. Five-sample self-consistency instead reduces balanced accuracy by 0.2123 and margin by 1.891, producing an audited-infeasible summary.

On Bash, the baseline is feasible at balanced accuracy 0.9345. PAR-D increases it to 0.9437 and improves the macro margin by 0.110, but the fold-based status becomes uncertain. Three other methods reduce balanced accuracy by 0.0238–0.0259 and also become uncertain. On TravelPlanner, rationale-feature scoring increases balanced accuracy by about 0.184 and the margin by 0.71, moving the baseline from audited-infeasible to uncertain.

Pool-specific baselines. A baseline voter is selected on macro FIT balanced accuracy in each rotation and then used across that pool's benchmarks. The main GPT winners are P_s, P_w, P_w, P_w, P_s. The cross-vendor baseline uses Gemini's B_s in three rotations and F_s in two. The rationale-feature baseline uses CRIT_s in four rotations and B_s in one.

Baseline status depends on the pool and scored population. Bash is feasible in the atom-level main GPT pool but audited-infeasible in both task-level pools. TravelPlanner is feasible in the main GPT pool but has an audited-infeasible baseline in the rationale-feature pool; it is not included in the cross-vendor pool.

What routing changes. Across 60 domain–rotation choices, PAR-D selects 11 of the 14 main GPT variants, whereas the matched baseline selects only P_s or P_w. Eight of the twelve domains retain one voter across all five rotations. Regex, SWE-bench-Verified, IFEval, and RAGTruth change voters. Re-deriving these choices from cached predictions using the FIT-only rule reproduces the reported PAR-D rates and margins.

Why a benchmark can have different margins across studies. Bash has $n = 523$, $\pi = 0.375$, $D = 5.39$, and baseline $M = +0.59$ in the main GPT scaling pool, compared with $n = 75$, $\pi = 0.733$, $D = 24.75$, and $M = -2.87$ in the held-out selection study. The demands differ by about 1.5 log units. Complexity's scaling baseline has $M = -0.31$, $\pi = 0.500$, and $D = 9.00$, whereas its selection-study values are $M = -1.81$, $\pi = 0.597$, and $D = 13.31$. TravelPlanner's scaling baseline has $M = +0.81$, $\pi = 0.041$, and $D = 0.39$, compared with $M = -0.64$, $\pi = 0.172$, and $D = 1.88$ in the selection study. These margins refer to different populations and procedures, not contradictory evaluations of the same fixed rule.

D.1.1 Rationale features and the external package

Rationale-feature scoring. The six-voter rationale pool asks whether fixed features of cached explanations add evidence beyond the votes. Its score achieves macro balanced accuracy 0.6886, compared with 0.6768 for the matched single verifier, a gain of 1.18 percentage points. Its benchmark statuses are $0/5/4$ in F/I/U order, with one status change.

The average gain hides substantial variation. Balanced accuracy decreases on six of nine benchmarks, by 0.0033 on Temporal to 0.0704 on Bash. The three gains are TravelPlanner ($+0.1844$), BFCL ($+0.1115$), and Complexity ($+0.0552$). On TravelPlanner, the score improves balanced accuracy from 0.5000 to 0.6844 and the margin from -0.64 to $+0.06$. This closes the point deficit but leaves the fold-based status uncertain. On Regex, the macro margin falls from -2.66 to -6.73 partly because one fold reaches the numerical log floor, not because that macro difference directly measures an evidence ratio (Appendix D.4).

Released external package. Of 1,756 package responses, 379 (21.6%) receive the fallback score because the parser cannot extract a structured value. The rates are 16.5% on instruction following, 22.5% on mathematical reasoning, 9.0% on function calling, and 43.3% on multi-hop question answering. They are similar across repetition indices: 21.4%, 19.8%, and 23.5%.

Four of the eight benchmark–setting pairs have an exactly matched baseline. Balanced accuracy and margin decrease in all four. On MATH-500, whose baseline balanced accuracy is 0.7645, the continuous score gives $\Delta\mathrm{BalAcc} = -0.1409$ and $\Delta M = -0.481$, with status U→I. Three judgments give -0.1271 and -0.273, with U→U. On BFCL, whose baseline is 0.7816, the two settings give changes of $(-0.1401, -0.289)$ and $(-0.1143, -0.207)$ in balanced accuracy and margin, respectively, both with I→I.

The two IFEval settings lack an exact-population match, and the two HotpotQA settings lack a baseline. They therefore do not enter the matched-transition count. The untested package paths and CARE are described in Appendix B.3.5; their absence should not be read as an empirical result about those methods.

D.2 Cross-vendor verification

This comparison asks whether a joint panel from several providers improves evidence relative to a single verifier selected from the same pool. The pool contains nine task-level voters: forward, critique, and backward styles from each of Claude Opus 4.7, Gemini 3.1 Pro Preview, and Grok 4.20. The panel averages all nine votes and selects its threshold on SEL. The baseline selects one of those nine voters by macro FIT balanced accuracy.

There is no GPT voter or separate within-vendor panel in this pool. The comparison is therefore between one joint cross-vendor panel and provider-blind single-voter selection, not between individual vendors.

Table 12: Joint nine-voter cross-vendor panel versus the single verifier selected from the same pool. n counts task-level records. Balanced accuracy and margin are macro averages over five rotations, and the status column reads baseline→panel. $\Delta\mathrm{BalAcc}$ is a difference in proportions; the macro $+0.0010$ corresponds to the rounded $+0.10$ percentage points in Table 1.

Benchmark	n	Base BalAcc	Panel BalAcc	ΔBalAcc (prop.)	Base M	Panel M	Status
Bash	100	0.5545	0.5059	-0.0486	-2.94	$-7.06^{\S}$	I→I
BFCL	139	0.7816	0.7563	-0.0253	-0.74	-0.72	I→I
Complexity	210	0.6971	0.6759	-0.0212	-1.82	-1.84	I→I
GSM8K	145	0.9679	0.9679	0.0000	$+0.48$	$+0.42$	U→U
IFEval	534	0.8228	0.8141	-0.0087	$+1.02$	$+0.91$	F→F
MATH-500	100	0.7645	0.8358	$+0.0713$	-0.75	-0.35	U→U
Regex	100	0.6576	0.6617	$+0.0041$	-3.00	$-6.91^{\S}$	I→I
Temporal	139	0.9346	0.9708	$+0.0362$	$+0.45$	$+0.65$	F→F
Macro (8)	—	0.7726	0.7735	$+0.0010$	—	—	2/4/2

§: one rotation accepts no correct output and its likelihood ratio is floored at 10^{-9} before taking logarithms. The macro margin is therefore partly determined by the numerical convention, not solely by a measured evidence change.

Panel result. The panel changes no benchmark status. Balanced accuracy improves on MATH-500 ($+0.0713$), Temporal ($+0.0362$), and Regex ($+0.0041$), is unchanged on GSM8K, and decreases on four benchmarks. The macro gain is 0.000975, or 0.0975 percentage points, reported as 0.10 percentage points in Table 1. Both baseline and panel have F/I/U counts of $2/4/2$.

MATH-500 has the largest balanced-accuracy gain and a margin increase of about 0.39 log units, but remains uncertain. This result concerns the specified panel and pool. It does not establish that providers are equivalent or that provider diversity cannot help.

Variation among the constituent voters. The nine judges can also be described individually on the 1,467 records carrying an entry for every voter. These raw-verdict summaries do not use held-out selection; their maximum is not a held-out estimate of a selected judge. They show heterogeneity rather than another audited method comparison.

Macro balanced accuracy ranges from 0.585 to 0.787. Gemini has the highest provider-level macro average under each style and Grok the lowest, but the best style differs: B_s for Claude and F_s for Gemini and Grok. The ordering also changes by benchmark. The three Grok judges lead on Bash, and six different judges are strongest across the eight benchmarks. Missing verdicts are excluded from these individual rates; Gemini's critique and backward styles account for 448 of the 614 missing judgments.

D.3 Size of the remaining evidence deficit

For a matched comparison with baseline margin $M_0 < 0$, the required improvement is $-M_0$ and the observed improvement is ΔM. Their ratio, $\Delta M/(-M_0)$, measures the fraction of the baseline log-evidence deficit closed by the alternative. These quantities use the fold-level scaling summaries in Appendix B.5.

Of the 36 comparisons with audited-infeasible baselines, 24 have no positive margin gain and 12 improve the margin. Among those 12 gains, the median closes 12.5% of the initial deficit. The median baseline deficit is 1.97 log units, corresponding to roughly a sevenfold increase in likelihood ratio to reach zero margin.

Only rationale-feature scoring on TravelPlanner closes a full point deficit, moving M from -0.64 to $+0.06$; the status remains uncertain. The next largest reported partial closures are 22% for rationale

Table 13: Improvement relative to the baseline evidence deficit. Gap closure is $\Delta M/(-M_0)$ for a negative baseline margin. Log-unit rows report minimum, median, and maximum. All quantities use the matched fold-level scaling analysis.

Quantity	Value
Comparisons on matched evaluation sets	117
of which the baseline is INFEAS.$^{\text{aud}}$	36 (34 offline, 2 external)
Required improvement $-M_{\text{baseline}}$	0.64 / 1.97 / 3.00
Comparisons with $\Delta M \leq 0$	24 of 36
Comparisons with $\Delta M > 0$	12 of 36
Observed ΔM where positive	0.02 / 0.25 / 0.71
Fraction of the gap closed, where $\Delta M > 0$	0.125 median, 1.098 max
Comparisons closing the full gap in point margin	1
Comparisons whose INFEAS.$^{\text{aud}}$ baseline becomes FEASIBLE	0

INFEAS.$^{\text{aud}}$ abbreviates INFEASIBLE-FOR-AUDITED-METHOD.

features on BFCL, 20% and 17% for weighted voting and naive Bayes in the SWE-bench-family scaling comparisons, and 16% for rationale features on Complexity. No audited-infeasible baseline becomes feasible.

D.4 NUMERICAL CONVENTIONS AND SENSITIVITY OF FOLD SUMMARIES

These post-hoc checks examine the numerical floor and the deterministic fold-vote summary. They supplement the prespecified effect sizes rather than replacing their cohort or inference procedure.

Which folds reach the log floor. Across 710 offline fold rows, three have zero held-out TPR and therefore reach the 10^{-9} likelihood-ratio floor: the cross-vendor panel on Bash in fold 0 and Regex in fold 4, and rationale-feature scoring on Regex in fold 4. All three are alternative-method folds with audited-infeasible matched baselines. None of their baseline folds is floored, and none belongs to a positive-gain comparison.

Zero TPR does not necessarily mean that a rule rejects everything. Two of these folds accept wrong outputs but no correct outputs; only the rationale-feature Regex fold accepts nothing. Separately, 181 of the 710 folds observe no false positive. Those folds remain finite under the add-one FPR convention and do not invoke the likelihood-ratio floor merely because their false-positive count is zero.

Recomputing macro margins with floors of 10^{-6}, 10^{-9}, and 10^{-12} preserves the same 34 offline audited-infeasible baselines and the same 12 positive gains. No baseline point margin becomes non-negative. Restricting the analysis to the 31 comparisons in which both methods accept a correct output in every fold gives the same qualitative conclusion. Because that restriction depends on the outcomes, it is a sensitivity analysis rather than a replacement cohort.

Sensitivity of the fold-vote label. Re-deriving labels from the stored fold statuses reproduces all 142 offline benchmark–method summaries. Of these, 29 are sensitive to a change in a single fold status; three lie within the 36-comparison audited-infeasible-baseline cohort. This is a property of the vote rule, not a significance test or an estimate of the false-transition rate. The scaling labels should therefore be read as summaries of the observed fold classifications.

E WHAT CALIBRATION DATA CAN SUPPORT

This appendix supports Section 4.4 by distinguishing empirical risk, a large-sample screen, and finite-sample support for the same fixed rules. Appendix B.6 gives the data allocation and estimators. Appendix E.1 reports the primary rules and the full design's outcomes; Appendix E.2 compares finite-sample procedures; and Appendix E.3 checks the coverage of the primary approximation. Table 2 reports the primary accepted-risk estimates, upper bounds, and diagnostic test risks.

E.1 PRIMARY RULES AND SENSITIVITY ACROSS CONFIGURATIONS

The primary analysis uses the eight vote-fraction rules fixed on FIT under rotation 4. Their task allocation is given in Appendix B.6. Table 14 reports the fitted thresholds, calibration counts, and decisions. Every procedure keeps the rule fixed and uses the same observed calibration-TPR requirement, $\beta = 0.50$.

Table 14: Fixed-rule calibration results. Panel (a) gives the primary rotation-4 vote-fraction rules and their calibration counts; panel (b) compares decisions at $\alpha = 0.10$ and $\beta = 0.50$. Primary split sizes are reported in Appendix B.6. HB denotes Hoeffding–Bentkus. No rule is rethresholded on CAL, and TEST does not change a decision.

(a) Fixed rule and calibration counts

Benchmark	Tasks G	Candidates K	τ	Accepts A	Errors E	TPR
GSM8K	43	6	0.1429	43	1	0.977
Temporal	39	5	0.7857	36	0	0.923
IFEval	195	16	0.0357	287	18	0.968
BFCL	51	4	0.0714	107	6	0.990
Bash	37	8	0.0714	127	7	0.984
Complexity	81	9	0.2857	54	4	0.617
MATH-500	32	8	0.1429	24	2	0.688
Regex	44	13	1.0000	75	9	0.569

(b) Primary decisions and screen passes across configurations

Benchmark	Empirical	Screen	HB	HB p	Screen passes / fixed rules
GSM8K	Pass	Pass	No support	0.669	15/15
Temporal	Pass	Pass	No support	0.580	15/15
IFEval	Pass	Pass	No support	0.657	6/15
BFCL	Pass	Pass	No support	0.850	9/15
Bash	Pass	Pass	Not evaluated	—	3/15
Complexity	Pass	No pass	No support	0.966	2/15
MATH-500	Pass	No pass	No support	0.993	2/15
Regex	No pass	No pass	Not evaluated	—	0/10

G counts parent tasks; A and E count scored outputs. All primary rules meet the observed TPR requirement. HB does not evaluate Bash or Regex under its task-size requirement. The final column counts large-sample screen passes among fixed-rule assessments; five additional Regex configurations have no eligible FIT threshold. The full design therefore contains 120 configurations, 115 fixed rules, and 52 screen passes.

Seven of the eight rules meet the empirical accepted-risk target, but only five pass the large-sample screen. Complexity and MATH-500 satisfy the point-risk criterion but not the upper-bound criterion. Regex fails both. The Hoeffding–Bentkus analysis supports none of its six applicable primary rules; the other finite-sample analyses are described in Appendix E.2.

Test outcomes do not revise these decisions. For example, Complexity has zero accepted errors on TEST, but that favorable realization does not change its earlier calibration result. The reported retention sensitivity also leaves the asymptotic-screen classifications unchanged when a confidence lower bound on TPR replaces the observed-TPR check.

Outcomes across the full design. The design contains 120 configurations: eight benchmarks, five rotations, and three score families. Five Regex configurations produce no eligible FIT threshold and therefore no rule to calibrate. Of the remaining 115 fixed-rule assessments, 52 pass the large-sample screen. Forty-nine passes have at least 30 calibration task clusters; the other three are BFCL's score families in rotation 1, each with 27 clusters. These counts describe repeated assessments on reused data, not independent replications or a simultaneous guarantee.

E.2 FINITE-SAMPLE SENSITIVITY

The finite-sample analyses provide no support for the rules they evaluate. This is not evidence that all those rules violate the target: a procedure can remain inconclusive because of limited sample size or because its required task-size structure is unavailable. The Hoeffding and Hoeffding–Bentkus analyses were prespecified. The empirical-Bernstein analysis was added after the primary results. Each uses the fixed rule and a pointwise confidence parameter of 0.05.

Hoeffding: a task-count limitation. The bound in Appendix B.6 has a necessary task-count condition. Since $\overline{Z} \geq -\alpha\overline{m}$ and

$$r \geq \overline{m}\sqrt{\frac{\log(1/\eta)}{2G}},$$

obtaining $\overline{Z} + r \leq 0$ requires

$$G \geq \left\lceil \frac{\log(1/\eta)}{2\alpha^2} \right\rceil = 150 \qquad (\alpha = 0.10,\ \eta = 0.05).$$

The cluster sizes cancel from this necessary lower bound. For a condition satisfying the procedure's assumptions, fewer than 150 calibration tasks preclude certification under this bound regardless of the acceptance outcomes.

Across the 115 configurations with a fixed rule, calibration task counts range from 27 to 227; 100 are below 150. This is a task-count diagnostic, not a claim that every finite-sample procedure is applicable to all 115 configurations. It excludes the five configurations without a fitted rule and does not remove the task-size restrictions in Appendix B.6.

The 15 configurations above the count threshold are all on IFEval, with 195–227 calibration tasks. Their observed values still do not satisfy the Hoeffding criterion: the radii are near 0.14, while $\overline{Z}$ ranges from -0.055 to $+0.016$. The reported sample-size comparison indicates that roughly an order of magnitude more clusters would be needed at these effect sizes. Checking the separate necessary condition $r \leq \alpha\overline{m}$ identifies the same 15 configurations.

Hoeffding–Bentkus: the same primary rules under a bounded-mean test. The Hoeffding–Bentkus analysis (Angelopoulos et al., 2025) uses the primary rotation-4 vote-fraction rules without rethresholding. It retains the accepted-risk target and the observed TPR requirement. Its implementation requires a common task-design-fixed bound on the number of scored outputs per task. Six primary benchmarks meet that requirement; none receives support (Table 14).

Bash and Regex are not evaluated by this implementation. Their atom counts come from the reference command or regex rather than the response, but vary over 1–11 and 1–8 atoms per task without a bound fixed by the benchmark design. The implementation does not replace that requirement with the observed sample maximum. These rows are unevaluated, not failed tests or demonstrated assumption violations.

Empirical Bernstein: a variance-adaptive comparison. We also evaluate the one-sided empirical-Bernstein bound of Maurer & Pontil (2009) on the transformed task-level quantity

$$X_t = \frac{E_t + \alpha(m_t - A_t)}{m_t} = \alpha + \frac{Z_t}{m_t} \in [0, 1].$$

When $m_t \equiv m$ is constant within a benchmark, $\mathbb{E}[X_t] \leq \alpha$ is equivalent to $\mathbb{E}[Z_t] \leq 0$, and hence to the accepted-risk target when expected acceptance is positive. With variable task sizes, this per-task normalization does not represent the same pooled target. The reported implementation therefore requires a constant number of scored outputs, not just a bounded number.

The five evaluated primary benchmarks are BFCL with four records per task and Complexity, GSM8K, MATH-500, and Temporal with two. Bash, Regex, and IFEval are outside this implementation's scope. IFEval has 1–3 verifiable instructions per task, determined by the source prompt rather than the response. All five evaluated rules retain their original FIT threshold, $\alpha = 0.10$, and observed TPR requirement; none is reselected on CAL.

The procedure supports none of the five. Its upper bounds on $\mathbb{E}[X_t]$ are 0.295 for GSM8K, 0.286 for Temporal, 0.277 for BFCL, 0.231 for Complexity, and 0.430 for MATH-500. The corresponding sample means lie between 0.054 and 0.094. Every bound exceeds $\alpha = 0.10$. These values are upper bounds on the transformed mean, not numerical upper bounds on accepted risk.

At the available 32–81 clusters, the additive term

$$\frac{7\log(2/\delta)}{3(G-1)}$$

is already between 0.11 and 0.28 for $\delta = 0.05$, exceeding the target before the variance term is added. This term alone requires at least 88 clusters to reach 0.10; that is necessary, not sufficient, for support under this procedure.

GSM8K, Temporal, and BFCL pass the asymptotic screen on the same calibration statistics. Their different finite-sample outcomes therefore reflect the procedures at these sample sizes, not different fitted rules or calibration samples. We did not evaluate additional threshold-selection or betting-based procedures, so these results do not establish that every valid finite-sample method would remain inconclusive.

E.3 Validation and limitations of the asymptotic screen

The primary screen assumes independent task clusters from a common super-population and uses a first-order large-sample approximation. Its nominal one-sided confidence level is 0.95, but the approximation does not ensure that level at the available sample sizes.

The reported coverage simulation evaluates 15 cluster-count–setting combinations, using $G \in \{10, 30, 60, 150, 500\}$ and 500 Monte Carlo replicates per combination. Observed coverage ranges from 0.856 to 1.000. Coverage below 0.95 occurs both below and above 30 clusters, so $G \geq 30$ is a sample-count descriptor, not a validated threshold for trusting the approximation.

The 52 passes are therefore reported as nominal large-sample screen passes. Neither the simulated coverage nor the finite-sample comparisons support treating them as deployment certificates.

F Audit Cases on Separate Data

The two cases illustrate different reasons not to proceed. The IFEval walkthrough reserves separate data for every stage. At $\alpha = 0.10$ it stops because Gate 1 is unresolved; at $\alpha = 0.20$ it completes both gates but receives no finite-sample support. AppWorld fixes a panel before its sealed evaluation, where the panel retains very few correct outputs. The protocols differ, and neither produces a finite-sample Gate-2 certificate.

Table 15: Audit case studies at $\beta = 0.50$, with $\alpha = 0.10$ except in the IFEval $\alpha = 0.20$ column. IFEval reports the selection-aware pilot result on one fixed allocation; AppWorld reports the fixed panel's pilot and sealed diagnostics. The cases use different protocols and neither supplies a finite-sample Gate-2 certificate.

	IFEval, $\alpha = 0.10$	IFEval, $\alpha = 0.20$	AppWorld pilot	AppWorld sealed
Allocated parent tasks	115 / 205 / 102 / 112	same	50	118
Scored records (correct / wrong)	107 (90 / 17)	same	50 (6 / 44)	118 (16 / 102)
Audited object	held-out search	held-out search	frozen panel	same panel
Decision	UNCERTAIN, MEASURE-MORE	FEASIBLE; Gate 2	diagnostic only	diagnostic only

IFEval's task allocation is 115 pilot, 205 FIT, 102 CAL, and 112 TEST tasks; its reported Gate-1 measurements use 107 pilot records, and both targets use the same pilot decisions. IFEval uses 2,000 parent-task bootstrap replicates with selection repeated. AppWorld resamples scenarios with demand fixed and floors the likelihood ratio at 10^{-9} before taking logarithms. All supplies use the add-one FPR convention.

F.1 TASK-DISJOINT IFEVAL WALKTHROUGH

A complete reservation of data roles. The component studies in Sections 4.2 and 4.4 reuse parent-task populations under different role assignments; their results cannot simply be chained into a sequential claim. This walkthrough instead allocates all 534 IFEval parent tasks to four disjoint roles:

$$|\mathcal{P}| = 115, \qquad |\mathcal{C}| = 419 = 205\ \text{FIT} + 102\ \text{CAL} + 112\ \text{TEST}.$$

The 115 pilot tasks are not additional to the 534, and the three reserved roles are subsets of $\mathcal{C}$. These counts differ from the component-study split in Appendix B.6.

The benchmark and score were chosen after the main analyses. IFEval and Temporal were the only two calibration benchmarks that were both feasible in the primary Gate-1 analysis and passes of the asymptotic screen. IFEval offered 534 parent tasks rather than Temporal's 100, failures identified by the official checker, and vote fraction as both its modal Gate-1 winner and planned Gate-2 score. The walkthrough is therefore illustrative, not prospective.

Before computing the walkthrough outcomes, a deterministic task-ID partition fixed the four disjoint sets. Within the pilot, five internal folds supported Gate-1 fitting, selection, and held-out evaluation over 26 candidate methods. The reported held-out statistics use 107 scored records: 90 correct and 17 wrong. The allocated pilot size and evaluated record count are therefore different quantities.

Outcome at $\alpha = 0.10$. The held-out supply is 1.65 against demand 1.70, giving $M = -0.030$ with bounds $[-0.720, +0.738]$. Algorithm 1 returns MEASURE-MORE. Gate 2 is not reached, even though its data roles were reserved; the walkthrough reports no algorithmic Gate-2 outcome.

A descriptive same-sample analysis of the source pilot gives supply 5.34, demand 1.78, TPR 0.593, and margin $+1.10$. This counterfactual does not affect the walkthrough decision and is not the controlled label-reuse experiment of Appendix C.2.

The case shows the loss of statistical resolution when data are reserved for separate stages. It does not establish IFEval infeasibility or overturn the full-population Gate-1 result. Only one fixed partition was evaluated.

Target sensitivity at $\alpha = 0.20$. Using the same allocation, we evaluate a second, less stringent target, $\alpha = 0.20$, with $\beta = 0.50$ and $\eta = 0.05$ unchanged. This retrospective sensitivity was specified in a written protocol after the $\alpha = 0.10$ result was known but before any $\alpha = 0.20$ outcome was computed. It exercises the complete data flow and does not replace the primary analysis. The split, Gate-1 family, search, and 2,000-replicate bootstrap are unchanged, and the $\alpha = 0.10$ Gate-1 result was reproduced exactly first. The protocol also fixed the Gate-2 hierarchy: Hoeffding is primary, Hoeffding–Bentkus is secondary, and the large-sample screen is diagnostic only.

Because α enters Gate 1 only through demand, supply and TPR are unchanged. Demand falls to 0.756, giving $M = +0.781$ with bounds $[+0.091, +1.549]$, and Gate 1 returns FEASIBLE.

On CAL (102 tasks, 168 outputs), the selected rule outputs, with 17 errors, for accepted risk 0.101 and TPR 1.000. The Hoeffding analysis gives $\overline{Z} = -0.163$ and $r = 0.215$, so $\overline{Z} + r = +0.052 > 0$ and provides no support. Hoeffding–Bentkus applies with the design bound of three outputs per task and gives $p = 0.299$, also providing no support. The large-sample screen gives $U = 0.143 \leq \alpha$, but it is diagnostic only. Because the observed risk is below α and the Hoeffding radius dominates, Algorithm 1 returns COLLECT-MORE-CALIBRATION. With the same $\overline{Z}$ and task sizes, about 178 calibration tasks would be needed. TEST (112 tasks, 164 outputs) was evaluated once after this action was fixed. The selected rule outputs, with 15 errors, for accepted risk 0.091.

F.2 APPWORLD PILOT AND SEALED TEST

Pilot and sealed outcomes. On the 50-task pilot, the panel accepts none of the six correct outputs. Its raw supply is zero and its floored margin is

$$\log(10^{-9}) - \log(66.0) = -24.913.$$

Floors from 10^{-12} to approximately 0.3 leave this margin negative.

On the 118-task sealed set, one backward-voter response is truncated and the fixed missing-vote rule withholds that task's panel decision. The panel accepts two outputs, both correct. Its supply is 12.875 and its margin is -1.494321; the critique voter has the same values. The reported one-sided margin bounds are $(-24.773, -0.677)$ under the scenario bootstrap of Appendix B.7.

What zero false positives do and do not show. At the observed TPR of 0.125, the add-one supply is

$$0.125\,(102 + 1) = 12.875.$$

This is the largest ratio the convention can report at that TPR and wrong-output count. In the bootstrap, the largest expressible margin is $\log(103) - \log(57.375) = +0.585$, while zero-true-positive replicates invoke the log floor and create the extreme lower tail.

The reported percentiles therefore describe a finite-data statistic affected by both conventions. Zero observed false positives leave the population FPR lower bound at zero, so these values do not provide a finite upper bound on the population likelihood ratio. The negative capped margin is diagnostic, not proof of population infeasibility.

The practical limitation is retention: the panel accepts only two of the 16 correct outputs, so its observed TPR is 0.125. This is below the paper's $\beta = 0.50$ target, although that comparison is post hoc because the frozen AppWorld rule omitted the TPR condition. Observing no accepted errors does not make a rule useful when it rejects nearly all correct outputs.

G Additional Related-Work Detail

Table 16 documents the evaluator-selection protocols discussed in Section 2. Some use the reported evaluation sample to choose an operating setting; others use a separate validation subset or cross-fitted folds. The table identifies the selected object and the scope of the held-out step. It is a targeted comparison of evaluation protocols, not a systematic review or an estimate of inflation in any published result.

Open evaluator models and instruction-following judge benchmarks motivate the judge-quality question (Kim et al., 2024b; Li et al., 2024; Zeng et al., 2024). Logit-free conformal prediction and prediction-based selection study additional guarantee settings (Su et al., 2024; Jin & Candès, 2023). Execution-based verification adds evidence from running a candidate or querying an environment (Chen et al., 2023; Ni et al., 2023; Gou et al., 2024); that differs from repeatedly judging the supplied record, as in the methods audited here.

Table 16: Examples of evaluator-selection protocols. Each row identifies the selected object and whether its reported evaluation uses the selection sample, a separate validation subset, or held-out folds. The scope is the specified selection step, not every stage of the cited method.

Work	Selected object	Selection data and evaluation
Auto-J evaluation, reward-model baselines (Li et al., 2024)	Reward-model tie threshold	The official leaderboard searches the best tie threshold over 0–2.0 in 0.01 increments while reporting agreement on the released pairwise set (58×24 pairs). Concerns those baselines, not Auto-J's training.
SFR-Judge (Wang et al., 2025)	Response ordering	Outside RewardBench, both A/B and B/A orders are evaluated and the better score reported; no test label enters training.
TutorBench (Srinivasa et al., 2025)	Judge model	Chosen for human agreement; agreement is reported on 250 human-rated examples from the 1,490-example benchmark.
ProfBench (Wang et al., 2026)	Judge model and reasoning-effort setting	Judge candidates are evaluated against human-labelled response–criterion pairs. The final judge is chosen using human agreement, bias, and evaluation cost, with reasoning effort adapted by domain/criterion type before use in report evaluation.
LEXam (Fan et al., 2026)	Judge ensemble	Expert agreement and judge comparisons use the same 50 question–answer pairs; overlap with the remaining benchmark is not specified.
Video-SafetyBench (Liu et al., 2025)	Judge threshold	Five-fold cross-validation: chosen on four folds and evaluated on the held-out fifth. Covers the threshold rather than every model choice.
Between Help and Harm (Arnaiz-Rodriguez et al., 2026)	Crisis-category classifier	Chosen against human labels on a 206-example validation subset, then used to annotate the test inputs.