ADMISSION WITHOUT ANSWERS: LABEL-FREE CERTIFICATION AND EXPERIENCE LEARNING FOR LLM-BASED OPTIMIZATION MODELING

Junbo Jacob Lian
Institute of Operations Research and Analytics
National University of Singapore, Singapore
Wenzhou Buyi Pharmacy, Wenzhou, China
jacoblian@u.northwestern.edu

Huiling Chen
College of Computer Science and Artificial Intelligence
Wenzhou University, Wenzhou, China
chenhuiling.jlu@gmail.com

Hanzhang Qin
Institute of Operations Research and Analytics
National University of Singapore, Singapore
hzqin@nus.edu.sg

Chung-Piaw Teo
Institute of Operations Research and Analytics
National University of Singapore, Singapore
bizteocp@nus.edu.sg

ABSTRACT

Experience-learning agents for optimization modeling improve by storing verified skills, but existing learners admit knowledge by checking against known answers, which real ticket streams do not provide. The natural label-free alternatives are unreliable: on a 300-problem label-blind stream, admitting every executable model poisons roughly one admission in four, while single-instance agreement accepts models that match at one value but differ elsewhere. We propose ADMITOR, an admission gate built on statistically calibrated external behavioral evidence. Candidates from three model families, prompting strategies, and solver stacks are run on instances resampled from an extracted parameter domain; agreement across the resulting value-function traces is summarized by a cross-family clique, and a calibrated threshold returns accept, abstain, or escalate. The preregistered false-discovery criterion holds on the calibration data; its stated transfer assumption fails on the wild stream, and we report this negative result in full, tracing most failures to benchmark texts that do not faithfully encode their labeled instances. Comparing four admission judges on a single collection of logs inside a state-of-the-art skill learner, ADMITOR raises admission precision to 0.927, against 0.871 for majority vote and 0.726 for execution success, yielding 3.1× and 8.0× fewer poisoned admissions. Its library is the smallest of the four and attains the highest macro accuracy across five public benchmarks, 58.4 against 54.8 for majority vote and 53.9 for the ground-truth-labeled library. The 3.5-point gain over majority vote is supported by a paired bootstrap and survives correction for a host-side anomaly. To our knowledge, ADMITOR is the first label-free admission mechanism designed around an explicitly calibrated false-discovery target, and the observed transfer failure identifies a necessary condition for extending it to wild streams. Code and data are available at https://github.com/junbolian/AdmitOR.

1 INTRODUCTION

Large language models can translate natural-language operational problems into executable optimization models. Modeling agents improve substantially when they *learn from experience* by distilling solved problems into reusable insights, skills, or exemplars (Kong et al., 2025; Yang et al., 2026; Liang et al., 2026). The obstacle is admission. Existing learners decide what enters their libraries by matching known optima (Kong et al., 2025), labeling trajectories against ground truth (Yang et al., 2026), or curating expert exemplars (Liang et al., 2026). Real ticket streams provide no answer book. We call this the *label wall*. Letting a model certify its own output does not solve

it: Xiu et al. (2026) admit on self-assessment and execution success, then report that naive retrieval from the resulting store *lowers* accuracy, consistent with poisoned memory and with prior evidence on unreliable self-correction and self-preference (Huang et al., 2024; Gou et al., 2024; Kamoi et al., 2024; Panickssery et al., 2024). On our label-blind stream of 300 problems, an execution-only criterion admits 878 candidate models, 241 of which disagree with the withheld answers.

A subtler failure arises when agreement is measured only at the original instance. Consider allocating 150 crates across three stores with capacities of 60, unit profits of 8, 6, and 4, and a contractual floor of 20 per store. The optimum is 960, and a model that *omits the floor* also returns 960 because the floor is slack at this instance. Single-point agreement, and therefore single-point majority voting, accepts both models. A check against a stored answer key admits both as well, being the same single-point test with the label in place of a second model: a correct answer at the stated instance does not certify a correct model. The disagreement appears only after the parameters change (Figure 1a). Answer keys are not infallible either. One widely used benchmark labels an instance as 200 even though its verifiable optimum is 250. Our pipeline therefore returned 250 and was scored as incorrect (Section 4.1).

ADMITOR (read *admitter*) takes a third route based on *external behavioral evidence, statistically calibrated*. Candidates from three model families under distinct prompting strategies and solver stacks are evaluated on instances resampled from a base-anchored parameter domain. Agreement is measured across *multi-instance value-function traces*. Under Proposition 1, inequivalent models disagree on a set of positive measure and the probability of missing that disagreement decays geometrically in the number of resampled instances. A maximum clique spanning at least two families summarizes the evidence, and calibration selects the admission threshold toward a nominal false-discovery target of $\alpha = 5\%$, with a finite-sample certified budget of 2α at a stated confidence (Proposition 2). Section 4.3 tests directly whether that calibration transfers to the wild stream. The gate returns ACCEPT with a certificate, ABSTAIN with a localized diagnosis, or ESCALATE. Only certified trajectories are distilled. Calibration rather than a fixed threshold is necessary because no sound, nontrivial fixed-threshold perturbation tester exists (Li & Hai, 2026).

Auditing a label-free judge still requires answer keys. We therefore evaluate ADMITOR on labeled public benchmarks before considering the unlabeled streams that motivate it. The gate replaces the admission judge inside a running skill learner (Yang et al., 2026). Under a collect-once, replay-many protocol, we generate candidates and solver logs once on the label-blind stream. Four judges then replay the same logs to build separate libraries using sealed-vault ground truth, majority vote, execution success, or ADMITOR. Admission precision increases from 0.726 to 0.871 and then to 0.927, with 241, 93, and 30 poisoned admissions, respectively. This gain comes from more frequent abstention, with an admission recall of 0.601. Across five public benchmarks, the gate-built library achieves the highest macro accuracy of the four, 58.35, while using the smallest admitted set. Its 3.5-point gain over majority vote remains positive under the prespecified sensitivity correction. The result also exceeds the 53.89 obtained by the ground-truth-labeled library. All comparisons are paired, intervals are obtained from a stratified bootstrap, and every decision criterion was preregistered before unblinding.

Contributions. **(i) Problem.** We identify label-free admission as the central barrier to deploying experience learners on real streams. We argue that the appropriate objective is not unattainable per-instance certainty (Section 5), but control of the false-discovery rate on the admitted stream. **(ii) Mechanism.** We introduce a cross-family panel, resampled value-function traces supported by an identifiability proposition, and finite-sample calibrated admission with a stated transfer assumption, a policy-matched calibration lemma, and a split-conformal extension at scale. To our knowledge, this is the first label-free admission mechanism to target an explicitly calibrated false-discovery rate. **(iii) Protocol.** We develop a collect-once, replay-many design that converts any collect-then-build learner into a controlled testbed for admission judges. Candidate generation is performed once, so every downstream difference is attributable to the judge. **(iv) Evidence.** ADMITOR produces $8\times$ fewer poisoned admissions than execution success and achieves the highest downstream accuracy of all four libraries with the fewest items. A paired bootstrap interval supports the gain over majority vote. **(v) Measurement.** Auditing every disagreement yields, to our knowledge, the first measurement of how often a widely used answer key is unreachable from its own problem text: at least 10.9% of admitted problems cannot be answered from the printed text by any method, which

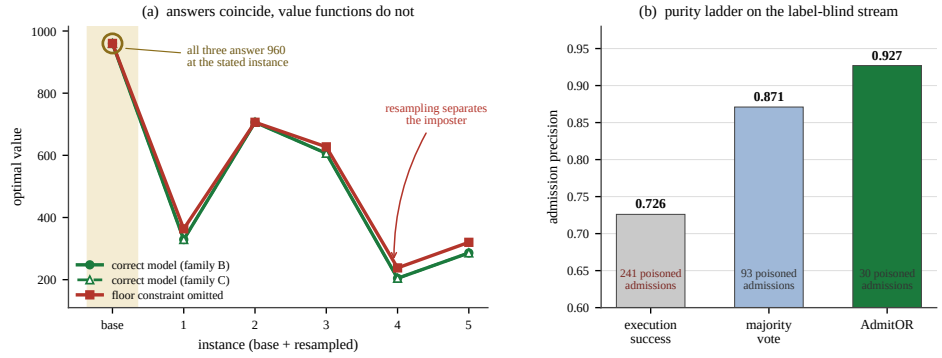

Figure 1: **(a)** The constructed running example executed through the released pipeline. Solver values and the clique verdict are produced by the code. A model that omits the floor constraint returns 960 at the stated instance, exactly as the correct models do, and separates from them on every resampled instance. The cross-family clique excludes it. **(b)** Admission precision of three label-free judges replaying the same 300-problem label-blind stream, measured against sealed reference labels. Annotations report the numbers of poisoned admissions.

lower-bounds the measurable false-discovery rate of every text-faithful system on this stream. We release the code, verdicts, run ledger, and the complete case packets behind that number.

2 Related Work

Experience learning with labeled admission. The systems most closely related to our setting learn reusable knowledge from solved problems, but all rely on answers for admission: insight libraries matched against known optima (Kong et al., 2025), skills distilled from ground-truth-labeled trajectories (Yang et al., 2026), and expert exemplar banks (Liang et al., 2026). AdmitOR replaces this supervision signal, evaluated inside one of these systems with the native ground-truth oracle as reference (Section 4.2).

Self-verification and its limits. Intrinsic self-correction can degrade reasoning (Huang et al., 2024), effective critique requires external tools (Gou et al., 2024), and models miss their own errors (Kamoi et al., 2024) while favoring their own outputs (Panickssery et al., 2024); Xiu et al. (2026) demonstrate the failure at the system level. Li & Hai (2026) prove that no sound and non-trivial fixed-threshold perturbation tester exists; they establish sound tests for individual models, whereas ours calibrates admission across multiple models.

Ensembles, voting, and juries. Vote- and jury-style aggregation (Liu, 2026), annotation-free routing (Niu et al., 2026; Guha et al., 2024), and cross-backbone debate (Lin et al., 2026) compare answers at single instances, which the running example shows to be insufficient, and none supplies an error budget for accepted outputs. The shared-error floor of Liu (2026) motivates our cross-family panel construction. Zhou (2026) isolates the mechanism behind that floor: a judge conditioned on a shown candidate scores plausibility rather than correctness, a strict three-judge ensemble still admits 55% of the resulting false positives, and the decisive variable is whether each judge commits an answer of its own before seeing the candidate. Our panel is de-anchored by construction, since each family generates and executes its own model before any comparison is made.

Verifiers for optimization modeling. Recent verifiers, semantic checkers, and behavioral test batteries (Liu et al., 2026; Rahman et al., 2026; Zhang et al., 2025; Li et al., 2026; Lian et al., 2026) filter individual generations, but none defines admission semantics for a persistent library, targets a calibrated FDR, or closes the experience-learning loop. We adopt the instrument of Lian et al. (2026) as a diagnostic layer.

Table 1: Properties targeted by each admission judge. *Label-free*: needs no answer key. *Behavioral*: tests the executable value function rather than text or self-report. *Beyond one point*: evidence spans resampled instances, escaping single-point coincidence. *Budget*: targets a calibrated false-discovery rate. Self-evaluation is listed for positioning only: it needs new model calls and so falls outside the collect-once, replay-many protocol of Section 4.2. We rely on the system-level evidence of Xiu et al. (2026) for its behavior.

Judge	Label-free	Behavioral	Beyond one point	Budget
Ground-truth labels	×	×	×	×
Execution success	✓	partial	×	×
Majority vote	✓	×	×	×
Self-evaluation	✓	×	×	×
ADMITOR	✓	✓	✓	✓

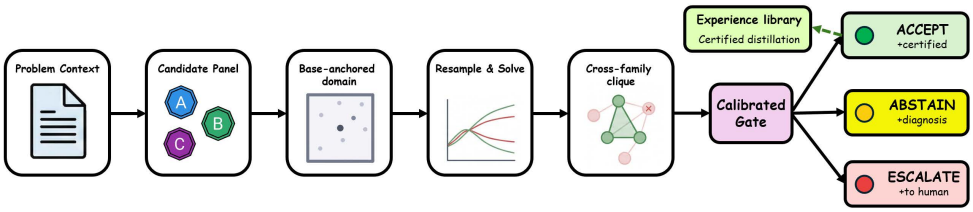

Figure 2: The ADMITOR gate. Candidates from three model families are evaluated on instances resampled from a base-anchored parameter domain. Agreement across sampled value-function traces is summarized as a cross-family clique. Finite-sample calibration converts clique evidence into a three-state decision toward a nominal false-discovery target of $\alpha = 5\%$, with a certified budget of 2α under the transfer assumption (Proposition 2). Certified trajectories are distilled into the experience library.

Conformal selection. Our admission layer is finite-sample calibrated selection toward an FDR target, with split-conformal Benjamini–Hochberg (Jin & Candès, 2023) as its at-scale form, applied to a new score, clique-structured cross-family agreement over resampled value functions, and hardened by policy-matched calibration: escalation ladders make admission adaptive, so calibration must use the ladder that deployment uses or the guarantee lapses. Selective prediction supplies the precision–coverage frame our three-state output makes explicit.

Table 1 summarizes four properties of an admission judge; to our knowledge, ADMITOR is the first label-free mechanism designed to address all four. Section 4.3 establishes the boundary: the calibrated budget does not transfer unchanged when deployment violates the assumptions of the calibration distribution.

3 THE ADMITOR GATE

Setup. A ticket is a natural-language problem text x specifying an optimization model with parameter vector $\theta \in \Theta$ and stated values θ_0. A candidate M is an executable program that builds and solves a model for any θ and returns an objective value. We denote its *value function* by $V_M(\theta)$ and call two candidates equivalent when their value functions agree on Θ. The gate receives a panel $\mathcal{C} = \{M_1, \ldots, M_k\}$ with family labels $f(M_i)$ and returns ACCEPT with a certified answer $\hat{z} = V(\theta_0)$, ABSTAIN, or ESCALATE. In the pilot, tickets without sufficient informative evidence are marked as UNINFORMATIVE, with escalation as the corresponding deployment action. The objective is to control the fraction of incorrect certifications among accepted admissions at level α. Figure 2 summarizes the five steps and Algorithm 1 in Appendix B gives the pseudocode.

Step 1: a decorrelated panel. We generate candidates from three model families under distinct prompting strategies and solver stacks. Each candidate uses a separate backend to reduce correlated errors. Every consensus mechanism has a shared-error floor, the probability that all panel members

make the *same* mistake; the floor is higher within one family because related models share training data and failure modes (Liu, 2026). Diversity across families and solver APIs lowers several sources of correlated failure but cannot detect a misinterpretation of the input text shared by all of them (Section 5).

Step 2: base-anchored domain extraction and resampling. An extractor LLM reads x and returns the stated base value and a relative or absolute perturbation domain for each parameter. The domain is anchored so that instance 0 is exactly the stated problem. Guardrails validate the specification. Conflicting ranges revert to relative perturbations, structural size parameters remain fixed so that resampled instances preserve their dimensions, and degenerate domains are widened with a warning. The design specifies *data-layer redundancy* at this stage: with E extractor families, panel entry requires agreement of the extracted key sets and base values within scoring tolerance, and disagreement routes the ticket to ESCALATE with a data-layer diagnosis (Appendix B.1). The pilot runs $E = 1$, and Section 4.3 measures the cost of that simplification. The harness then draws $\theta_1, \ldots, \theta_m$ independently from the domain and evaluates every candidate on every instance, recording the objective value and solver status for each run. An instance is marked as uninformative and excluded when too few candidates return comparable values because of infeasibility or execution failure. The gate requires a minimum number of informative instances before issuing a verdict.

Step 3: value-function consensus. Two candidates agree on an instance when their objective values match within a relative tolerance. They are empirically consistent when they agree on every informative instance, including the base instance. Pairwise consistency defines a graph over $\mathcal{C}$. Positive evidence is given by a maximum clique that spans at least two model families. The common clique value at θ_0 becomes the certified answer. Each candidate outside the clique is rejected with an instance-level diagnosis that identifies a resampled instance on which the candidate differs. This evidence localizes the modeling disagreement. In the running example, a negative unit profit activates the omitted floor constraint. The gate therefore replaces single-point agreement with agreement across a sampled value-function trace. The following result characterizes the corresponding detection probability:

Proposition 1. *For linear and mixed-integer models over a full-dimensional perturbation domain, the candidate value functions are piecewise polynomial in θ. They are affine in right-hand sides when costs are fixed and affine in costs when right-hand sides are fixed. If two candidates are not equivalent on the domain, the disagreement region contains an open set. Under continuous resampling, the probability that m independent instances all miss this region decreases geometrically with m. The rate depends on the probability measure of the disagreement region. A formal statement and proof appear in Appendix A.*

Step 4: finite-sample calibrated admission. Clique evidence is summarized by a score. The deployed score is the lexicographic triple of family coverage, clique size, and the number of informative instances, encoded as a scalar (Algorithm 1). It therefore takes finitely many values, four of which are attained in the pilot. Pairwise agreement margins are recorded but do not enter the deployed score. Section 4.3 reports the consequence of this choice. We calibrate the acceptance threshold τ on problems with ground truth verified *by construction*: the unchanged gate evaluates a stratified sample of solver-verified synthetic instances, so every accepted certificate can be labeled true or false without benchmark data. A fixed threshold is not a valid substitute, because no sound and nontrivial fixed-threshold perturbation tester exists (Li & Hai, 2026); nor is picking the threshold by its empirical calibration error alone, which carries no statement about deployment. What licenses the selection is the following pair.

Assumption 1 (Transfer)**.** *(i)* Faithful encoding*: the problem text determines the labeled instance, so the extracted base specification identifies it up to scoring tolerance. (ii)* Certificate exchangeability*: accepted deployment certificates with score at least τ are exchangeable with calibration certificates at the same score with respect to the true/false label. Clause (i) holds on the calibration set by construction.*

Proposition 2 (Calibrated admission)**.** *Let T be the attainable score grid ($|T| = 4$ in the pilot), p_τ the false-certificate probability among accepted certificates with score at least τ, and U_τ the level-$(1-\delta)$ Clopper–Pearson upper bound from the calibration counts. The fit selects $\tau^* = \min\{\tau \in T : \hat{p}_\tau \le \alpha,\ U_\tau \le 2\alpha\}$. Then (i) for each fixed τ, $\Pr(p_\tau \le U_\tau) \ge 1 - \delta$ exactly; (ii) over the data-dependent selection, $p_{\tau^*} \le 2\alpha$ holds simultaneously with probability at least $1 - |T|\delta$; (iii) under*

Assumption 1, the same bounds apply to the false-certification rate among deployment admissions at τ^*; and (iv) as the calibration null count grows, Benjamini–Hochberg over split-conformal p-values replaces the grid certificate and controls the realized FDR at level α in finite samples under clause (ii) alone (Jin & Candès, 2023). Proof in Appendix A.

Throughout, $\alpha = 5\%$ is the nominal selection target applied to $\hat{p}_\tau$; the finite-sample *certified* budget is $2\alpha = 10\%$, and the two figures should not be conflated. The pilot instantiates $\delta = 0.05$ per threshold: the selected-threshold certificate holds at 95% read per threshold, the simultaneous statement of Proposition 2(ii) at $1 - |T|\delta = 80\%$ over the four-point grid, and the 95%-simultaneous reading ($\delta = 0.0125$) is recomputable from the released calibration counts. The guarantee is conditional and testable: Section 4.3 runs the fitted rule on the wild stream and measures the violation of Assumption 1. Deployment imposes one additional requirement:

Lemma 1 (Policy-matched calibration, informal)**.** *If admission is adaptive, borderline cases enter an escalation ladder that adds instances or candidates before a new decision. The guarantee then holds only when calibration uses the same ladder policy as deployment; calibration on single-pass decisions does not remain valid under escalated deployment (formal statement and proof in Appendix A).*

Step 5: certified distillation. Only ACCEPT verdicts contribute to learning: each admitted trajectory is relabeled with the certified value and passed to the unmodified distillation procedure of the host, so the library grows as it would under a ground-truth oracle with a certificate replacing the missing answer. ABSTAIN and ESCALATE tickets are excluded; Section 4.2 quantifies the trade-off.

What the gate cannot do. The gate certifies agreement among independently derived behaviors, not ground-truth intent: a misinterpretation shared by every family is invisible to any amount of resampling, the statistical target is the false-discovery rate of the admitted stream rather than the correctness of each admission, and higher precision costs abstention (Section 5).

4 EXPERIMENTS

Setup and pins. The host experience learner (Yang et al., 2026) runs the released pipeline with a single fixed backbone, DeepSeek-V3.2 in non-thinking mode at temperature 0. The only modifications to the host are failure handling and call logging, both documented in the release. The panel of the gate uses the three model families introduced in Section 3: DeepSeek-V3.2, GPT-5.4, and Claude Sonnet 4.6. We generate one candidate per family at temperature 0. The first and third families use direct modeling, whereas the second uses a structured strategy. Solver stacks are divided between Pyomo with HiGHS and `gurobipy`. Each certification includes the mandatory base instance and $m = 5$ resampled instances generated with a fixed seed. All model versions, prompts, tolerances, and solver builds are fixed in the released configuration. Certification runs are independent of the generation runs of the host and are reported separately in the run ledger (Appendix C).

4.1 HOST REPRODUCTION AND THE MEASURING STICK

Before replacing the admission judge, we reproduce the complete host on five public benchmarks using the released skill library and a uniform scorer because the repository provides no scoring code. The scoring rule appears in Appendix B, and per-benchmark differences appear in Appendix C. The reproduced macro accuracy is within 2.6 points of the reported result. This stage yields two findings that shape the subsequent analysis. First, the evaluation labels are imperfect. On ComplexOR, our pipeline returns 250 for an instance labeled 200, and manual verification establishes 250 as the true optimum. The model is therefore penalized for a correct answer. ComplexOR scores 66.7 under the published labels and exactly matches the reported 72.2 after correcting that label. The protocol was fixed before any judge was evaluated, so the main results retain the published labels and label errors count against every method, including ADMITOR. Second, the OptMATH reproduction has a known deficit of 3 to 5 points. Partial reruns attribute this difference to both solver contention and a genuine capability gap. We annotate the absolute scores and base no central claim on them.

Table 2: Downstream accuracy (%, round-aware scorer) of the host when equipped with the library produced by each judge, evaluated on five public benchmarks. Every arm replays the same collection logs, and the judge is the only difference. *Macro* averages the five benchmarks equally, while *Micro* weights by item over all 1100 items. Bold marks the best result in each column.

Judge	ComplexOR	IndustryOR	Mamo.C	OptMATH	OptiBench	Macro	Micro
Ground truth	**66.67**	31.00	52.61	**56.02**	63.14	53.89	57.18
Execution success	**66.67**	35.00	53.08	55.42	**72.40**	56.51	62.64
Majority vote	61.11	33.00	53.55	54.22	72.23	54.82	62.18
ADMITOR	**66.67**	**39.00**	**57.82**	**56.02**	72.23	**58.35**	**63.91**

4.2 THE JUDGE SWAP

For the 300-problem label-blind stream, we collect candidate generations and solver logs once. Four judges then replay the same logs, assign labels, and pass their admitted trajectories to the unmodified distillation procedure of the host. This process produces four libraries that differ only in the judge. Certification returns 174 ACCEPT, 114 UNINFORMATIVE, 10 ABSTAIN, and 2 error outcomes. The two negative outcomes have different causes. ABSTAIN indicates value-level disagreement and occurs 10 times. In contrast, none of the 114 UNINFORMATIVE cases arises from conflicting evidence. Of these cases, 65 are dominated by infeasible resampled instances, 45 contain at least two candidates that never return a value, and 4 result from execution failure. Coverage is therefore limited by the harness as much as by the evidence standard. Figure 1b reports admission precision against the sealed vault over admitted *candidate models*, which are the units on which each judge acts: 878 candidates for execution success, 721 for majority vote, and 413 for ADMITOR. Admission recall is the proportion of correct executable candidates recovered, so execution success attains 1.0 by construction. Section 4.3 instead counts problem-level certificates, of which the calibrated rule admits 138. The two units are not interchangeable. Library sizes fall as the admission rules become more selective: 163 files for execution success, 145 for majority vote, 130 for ground truth, and 101 for ADMITOR.

Table 2 answers the first two preregistered questions. The absence of labels imposes no downstream ceiling. Every label-free arm exceeds 70% of the ground-truth macro accuracy, and all three also exceed the observed ground-truth result. Higher admission precision produces measurable gains. ADMITOR matches or exceeds majority vote on all five benchmarks and adds 3.5 macro points while using the smallest library. Figure 4 in Appendix C plots it.

Intervals come from a paired stratified bootstrap on the macro scale with 10,000 resamples drawn within each benchmark. Against majority vote, the gain is $+3.53$ points, with a 95% interval of $[+0.87, +6.75]$. Against the ground-truth arm, the gain is $+4.46$ points, with an interval of $[+2.37, +6.64]$. Because the ground-truth arm contains the host-side anomaly described below, we repeat both comparisons on the item set that the sensitivity analysis defines, removing the 85 affected items from every arm so the comparison stays paired. The gain over majority vote is unchanged at $+3.49$ points, $[+0.82, +6.69]$. The gain over the ground-truth arm narrows to $+2.14$ points, $[+0.05, +4.29]$. The certified library therefore stays ahead of every alternative on both bases, and the preregistered criterion K2, which concerns majority vote, is met with essentially the same margin in both analyses. The benchmarks differ in size by a factor of thirty. The item-weighted column of Table 2 preserves the ordering with smaller margins, and on ComplexOR a single item is worth 1.11 macro points.

Two qualifications are necessary. First, the OptiBench result of the ground-truth arm contains an anomaly. For 85 of 605 items, the failure occurs during *skill selection* rather than optimization modeling. The selector of the host repeatedly generates an identifier that is not present in the library. Following the protocol, these cases count as errors, as do analogous failures in every other arm. Each affected row is identified in the released logs. Excluding these cases in a sensitivity analysis raises ground-truth accuracy to 73.5 on OptiBench and 55.95 in macro average, which remains below the result of ADMITOR. The corrected intervals above use this same item set after removing the 85 affected items from every arm. This anomaly shows that library *contents* can affect the reliability of the selection stage. Admission policy determines both what a library contains and how reliably the library can be searched. Second, the execution-success arm performs well downstream despite

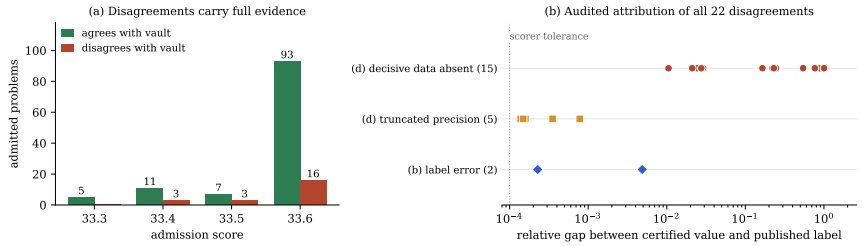

Figure 3: The anatomy of the K3 failure, from the released replay report. (a) Admitted problems by admission score: sixteen of the twenty-two disagreements sit at the maximal score (three families, three-member clique, all six instances informative), indistinguishable from the 93 correct admissions there, so no clique-geometry threshold separates them. (b) Every disagreement placed by its relative gap to the published label and colored by the audited root cause: the five truncated-precision cases hug the scorer tolerance, the two label errors sit among them, and the fifteen cases whose decisive data never appear in the text spread across two orders of magnitude up to sign flips.

its poisoning rate because, at this scale, a larger library partly compensates for lower precision. Our claim is not that permissive admission fails. Instead, certified admission achieves higher accuracy with approximately one-third fewer items and provides an explicit audit trail. Section 4.3 asks whether the targeted false-discovery rate transfers beyond calibration.

The family-by-outcome matrix (K4). The final preregistered check asks whether the three arms have distinct failure profiles; near-identical profiles would weaken the motivation for consensus. Table 3 in Appendix C reports the matrix mined from the stored artifacts, and the profiles differ significantly along interpretable dimensions: execution failures track the solver stack (the two Pyomo arms fail on 15 to 20% of problems against 8% for Gurobi, reproducing the development-time pattern at scale), instance infeasibility is flat across arms at 71 to 75 as expected for a property of the perturbed instance, and the arm with the fewest execution failures reaches value comparison most often, producing 30 value-level disagreements against 9. Shared instances make the profiles more similar under the null of identical mechanisms, so the observed separation is conservative. K4 meets the criterion, with two bounds on its reading: model family, prompting strategy, and solver stack co-vary by design, so the check establishes that the arms are not interchangeable rather than isolating family as the source of decorrelation, and the evidence concerns decorrelation conditional on a shared extracted specification, not at extraction itself (Section 4.3).

4.3 Calibrated admission: the criterion that fails

We fit the calibrated admission layer according to Section 3. The unchanged gate evaluates a stratified sample of 150 solver-verified NANO-CO instances (Yang et al., 2026). Calibration and deployment run the identical single-pass policy, including the one base-instance repair round of Algorithm 1, so the requirement of Lemma 1 is met by construction in this pilot. The gate returns 103 ACCEPT, 36 UNINFORMATIVE, 9 ABSTAIN, and 2 error outcomes. Comparison with the verified answers labels the accepted certificates as 93 true cliques and 10 false cliques, corresponding to a raw false rate of 9.7%. The smallest threshold for which the selected calibration set satisfies the criterion of Proposition 2 is $\tau = 33.3$. Under the deployed score this threshold is equivalent to requiring a unanimous three-family, three-member clique with at least three informative instances. Every admitted certificate attains the maximum on family coverage and clique size, so the attainable scores differ only in the number of informative instances. The selected calibration subset meets the target. We then replay this rule on the 170 non-development certification verdicts from Section 4.2, without additional model calls. The rule admits 138 cases, or 81.2% of the accepted verdicts. We measure the realized false-discovery rate out of sample against the sealed vault.

The preregistered criterion fails and we report it without modification (Table 5 and Appendix C). Under the uniform scorer, 22 of 138 admitted certificates disagree with the sealed vault, a realized FDR of 15.9% with a 95% upper bound of 22.0%. The stricter admission-time rule finds 26 disagreements, or 18.8%. The distinction between the nominal target and the certified budget does not soften this verdict: the realized rate exceeds the 5% target, and its upper bound exceeds the

$2\alpha = 10\%$ certified budget, so the preregistered criterion fails on both readings. Tightening as the protocol permits does not recover the target. The maximum score, requiring three families, a three-member clique and six informative instances, still admits 109 cases with 16 disagreements (14.7%), and across the four attainable thresholds the realized values are 15.9%, 16.5%, 16.0% and 14.7%. Those thresholds differ only in the number of informative instances demanded, the other two components already being saturated, so the flat profile shows that insufficient clique evidence is not the cause. Sixteen of the twenty-two carry *full* evidence, all three families agreeing on every sampled instance, so at the maximal score they are indistinguishable from the 93 correct admissions and no clique-geometry threshold can separate them. Additional resampling from the same extracted specification may reveal further model-level differences, but it cannot establish whether that specification faithfully represents the omitted source data. The audit below locates the observed defect at this upstream boundary rather than in the certified programs.

We audit all twenty-two using case-specific review packets that contain the problem text, the code of every clique member, both values, the relative gap, and the clique geometry. The protocol assumes neither value correct and requires each verdict to cite a span of the text with a code line or an explicit derivation. Three cases were settled by exact re-derivation rather than judgment. The audit is the authors' own, which bounds the strength of the attribution. The released packets enable independent re-audit. The results overturn our initial taxonomy (Table 4 in Appendix C and Figure 3b). We expected to find gate defects, in which the clique omitted a stated constraint, and legitimate alternative interpretations of ambiguous text. The audit finds no case in either category. Two cases are direct label errors. We independently re-derive both certified values without using the gate. For a six-month capacitated lot-sizing problem in which the capacities never bind, exact dynamic programming reproduces the certified optimum. For a contract-allocation problem whose complete instance is printed, an exact solution of the integer program produces the same result. In both cases, the published label is *below* the true minimum of the stated minimization problem and is therefore unattainable by any feasible solution.

The remaining twenty cases share a root cause that was not preregistered: *the problem text is not a faithful encoding of the labeled instance*. In fifteen cases, the printed text omits decisive data, typically a cost or demand matrix. In five cases, parameters are printed at insufficient precision, so the optimum of the stated text differs from the label by more than the scoring tolerance. One example can be diagnosed from the sign of the objective alone. A six-producer allocation problem prints two cost entries and states that the remaining entries have similar structures. The certified objective is *negative*, which is impossible under any nonnegative cost matrix. The extractor must therefore have supplied values that were not present in the text, while the magnitude of the label is consistent with a real instance of the stated form. We call this category (d): the text is not a sufficient statistic for the label.

Category (d) is a property of the benchmark, not a solver failure. At least fifteen of the 138 admitted problems, 10.9%, cannot be answered from the printed text by any method. The rate is measured on the admitted subset, where a false-discovery rate is defined, and it lower-bounds the measurable FDR of any text-faithful system whose admitted set resembles ours. This estimate is conservative because we audited only the disagreements, so a problem for which fabricated values happen to reproduce the label remains undetected among the 116 agreements, and the count also excludes the five truncated-precision cases. Synthetic suites of unanswerable problems test whether a model detects missing data; to our knowledge, no prior work has measured this failure mode inside the answer key of a widely used real benchmark or quantified the resulting FDR floor. The released case packets make this reproducible on other streams.

These cases mark the precise boundary of the decorrelation argument. All three families solve a common base specification produced by one extractor, so when the text omits a matrix the extractor fills it and the three independent families agree consistently on the same fabricated instance. Cross-family consensus certifies the extracted instance, not the one the benchmark author intended, because downstream agreement cannot detect an upstream error on which every candidate is conditioned. The remedy is architectural rather than a stricter threshold: one extractor per family, with data-layer agreement required before model-level admission, the stage specified in Appendix B.1 and run with $E = 1$ in this pilot. A post-hoc screen from text-derivable bounds (the sign of the optimum, cost lower bounds implied by total demand) flags nine of the twenty-two disagreements without labels; it suggests an inexpensive pre-consensus stage and affects no reported result.

Two conclusions extend beyond this pilot. NANO-CO texts are generated from their instances, so clause (i) of Assumption 1 holds there by construction, and the wild stream violates exactly that clause: the criterion's failure is a measured violation of a stated assumption, localized by the audit, not a defect in the concentration argument of Proposition 2. Moreover, resampling cannot validate the input it is conditioned on: perturbing an extracted specification separates incorrect models of that specification without testing whether it matches the text, so a valid error budget on wild streams needs data-layer redundancy before model-agreement analysis, escalating high-evidence cases.

Appendix C works through the 200-versus-250 label-error instance at admission and at evaluation, and a calibration case where models differing by an entire constraint family sit 0.023% apart. Appendix B reports the preregistered criteria, scoring rule, verbatim prompts, and a certified skill; Appendix C gives the reproduction details and run ledger. Certification spends a fixed budget per problem: one extraction call, one generation call per family, at most one repair call per candidate, and $(m + 1)k$ solver runs ($m = 5$, $k = 3$ here). Majority vote over the same panel consumes the generation calls alone, so the gate adds one extraction call per problem and six solver runs per candidate, plus any repair calls. Complete call-level logs accompany the release.

5 LIMITATIONS

The gate certifies agreement across independently derived behaviors, not the intended meaning of the problem: if every family reads the text the same incorrect way, their value functions agree on every sampled instance and resampling cannot detect it. The limitation belongs to the observation channel, which is why the objective is the admitted-stream FDR rather than a per-admission guarantee.

Second, higher admission precision reduces coverage: the gate emits nothing on a substantial fraction of the stream, mostly for want of informative instances rather than through disagreement (Section 4.2). The trade-off is appropriate when a false admission can poison a persistent library and costly when every ticket requires an answer; escalation reserves human attention here.

Third, the guarantee of Proposition 2 is conditional on Assumption 1, and Section 4.3 measures its violation: false certifications carrying complete clique evidence survive every threshold based on clique geometry alone, and deploying an escalation policy other than the calibrated one weakens the guarantee further (Lemma 1). All families share one extracted base specification at $E = 1$, so consensus cannot detect extraction errors and each certificate is conditional on the extracted instance.

Fourth, the host uses one generation backbone, so the judge swap isolates the admission judge without testing backbone diversity in the host; the panel is cross-family and extends to multi-backbone hosts.

6 CONCLUSION

ADMITOR replaces answer-based admission with externally generated behavioral evidence: three model families are evaluated across resampled value-function traces, agreement is summarized by a cross-family clique, and calibration maps it to a false-discovery target with explicit abstention. Because the collect-once, replay-many protocol changes only the judge, differences among the libraries are attributable to the judge: the certified library achieves the highest precision and downstream accuracy with the fewest items, its gain over majority vote supported by a paired bootstrap interval positive under sensitivity correction. Answer keys are not a precondition for downstream performance.

The evaluation fixes the boundary of the claim: the calibrated target does not transfer to the wild stream, where the audit attributes the dominant failure to benchmark texts that do not faithfully encode their labeled instances, violating the stated transfer assumption. ADMITOR supplies the framework, comparison, and audit trail for the data-layer stage that wild-stream certification requires.

REFERENCES

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 (ICLR), 2024.

Neel Guha, Mayee F. Chen, Trevor Chow, Ishan S. Khare, and Christopher Ré. Smoothie: Label free language model routing. *Advances in Neural Information Processing Systems*, 37:127645–127672, 2024.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations (ICLR)*, 2024.

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

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can LLMs actually correct their own mistakes? A critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12, 2024.

Minwei Kong, Ao Qu, Xiaotong Guo, Wenbin Ouyang, Chonghe Jiang, Han Zheng, Yining Ma, et al. AlphaOPT: Formulating optimization programs with self-improving LLM experience library. *arXiv preprint arXiv:2510.18428*, 2025.

Haifeng Li and Mo Hai. Falsification-based verification of LLM-generated optimization models: Sound test batteries and their detection limits. *arXiv preprint arXiv:2607.16646*, 2026.

Zhong Li, Zihan Guo, Xiaohan Lu, Juntao Wang, Jie Song, Chao Shen, Jiageng Wu, and Mingyang Sun. OptArgus: A multi-agent system to detect hallucinations in LLM-based optimization modeling. *arXiv preprint arXiv:2605.11738*, 2026.

Junbo Jacob Lian, Yujun Sun, Huiling Chen, Chaoyu Zhang, Hanzhang Qin, and Chung-Piaw Teo. ReLoop: Structured modeling and behavioral verification for reliable LLM-based optimization. *arXiv preprint arXiv:2602.15983*, 2026.

Kuo Liang, Yuhang Lu, Jianming Mao, Shuyi Sun, Chunwei Yang, Congcong Zeng, Xiao Jin, Hanzhang Qin, Ruihao Zhu, and Chung-Piaw Teo. Large-scale optimization model auto-formulation: Harnessing LLM flexibility via structured workflow. *arXiv preprint arXiv:2601.09635*, 2026.

Jianghao Lin, Zi Ling, Chenyu Zhou, Tianyi Xu, Ruoqing Jiang, Zizhuo Wang, and Dongdong Ge. From soliloquy to agora: Memory-enhanced LLM agents with decentralized debate for optimization modeling. *arXiv preprint arXiv:2604.25847*, 2026.

Haoyang Liu, Jie Wang, Boxuan Niu, Xiongwei Han, Yian Xu, Mingxuan Ye, Zijie Geng, et al. Opt-Verifier: Unleashing the power of LLMs for optimization modeling via dual-side verification. *arXiv preprint arXiv:2605.29556*, 2026.

Ning Liu. LLMs as a jury: Cross-model consensus can outperform process reward models for LLM reasoning. *arXiv preprint arXiv:2607.10139*, 2026.

Tianyi Niu, Justin Chen, Genta Indra Winata, Shi-Xiong Zhang, Supriyo Chakraborty, Sambit Sahu, Yue Zhang, Elias Stengel-Eskin, and Mohit Bansal. Routing with generated data: Annotation-free LLM skill estimation and expert selection. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32441–32466, 2026.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems*, 2024.

Sumaya Abdul Rahman, Seckhen Ariel Andrade Cuellar, Ghani Raissov, and Mohammad Raza. VeriSimpl: Robust optimization modeling from natural language using simplification-based verification. *arXiv preprint arXiv:2607.20474*, 2026.

Xianchao Xiu, Jianhao Li, Huangyue Chen, and Wanquan Liu. OptGraph: Large language models enhanced evolutionary optimization via graph retrieval-augmented generation. *arXiv preprint arXiv:2607.27918*, 2026.

Haochen Yang, Ke Zhao, Mengyuan Ma, Xingyu Lu, Xiangfeng Wang, and Hong Qian. OptSkills: Learning generalizable optimization skills from problem archetypes via cluster-based distillation. *arXiv preprint arXiv:2605.29829*, 2026.

Yansen Zhang, Qingcan Kang, Yujie Chen, Yufei Wang, Xiongwei Han, Tao Zhong, Mingxuan Yuan, and Chen Ma. SAC-Opt: Semantic anchors for iterative correction in optimization modeling. *arXiv preprint arXiv:2510.05115*, 2025.

Chenyu Zhou. More convincing, not more correct: Self-play reward hacking of reference-free LLM judges. *arXiv preprint arXiv:2607.05904*, 2026.

A PROOFS

A.1 IDENTIFIABILITY BY RESAMPLING

Assumptions. **(A1)** The perturbation domain $\Theta \subset \mathbb{R}^d$ is compact and full-dimensional, and the resampling law P has a density bounded below by $\rho > 0$ times Lebesgue measure λ on Θ. **(A2)** Each candidate M is solver-exact for the model its code constructs. On input θ, it returns $V_M(\theta) = \min_{x \in X_M} c_M(\theta)^\top x$ subject to $A_M x \le b_M(\theta)$, where $c_M(\cdot)$ and $b_M(\cdot)$ are affine in θ, the constraint matrix A_M does not depend on θ, and X_M imposes integrality on a subset of coordinates. **(A3)** Over Θ, each candidate has finitely many optimal integer patterns. Bounded integer variables and rational data guarantee this condition. In addition, V_M is finite on Θ. The harness discards infeasible or unbounded draws as uninformative, so each verdict is conditional on finiteness.

Lemma 2 (Piecewise-polynomial structure)**.** *Under (A2) and (A3) there is a finite partition of Θ into relatively open, full-dimensional cells $C_1, \dots, C_K$ (plus a Lebesgue-null boundary set) such that on each cell V_M coincides with a polynomial of degree at most two in θ, affine in θ whenever the perturbation touches only b_M or only c_M.*

Proof. Fix an integer pattern z and consider the LP obtained by fixing the integer coordinates at z. Parametric linear programming partitions the parameter space into finitely many polyhedral critical regions. Within each region, an optimal basis B remains fixed, and the LP value is $c_M(\theta)^\top x_B(\theta)$. Because $x_B(\theta)$ is affine in $b_M(\theta)$, it is also affine in θ. Since $c_M(\theta)$ is affine in θ, their product is a polynomial of degree at most two. It reduces to an affine function when only one block varies. By (A3), the MILP value is the pointwise minimum of finitely many such functions, one for each integer pattern feasible at θ. Pattern feasibility is a polyhedral condition because it is affine in $b_M(\theta)$. A common refinement of the critical regions and the loci at which the minimizing pattern changes yields the finite cell partition after excluding the null set of cell boundaries. □

Proposition 3 (Identifiability by resampling, formal)**.** *Let M_1, M_2 satisfy (A1) through (A3), write $\Delta(\theta) = V_{M_1}(\theta) - V_{M_2}(\theta)$, and let $D = \{\theta \in \Theta : \Delta(\theta) \neq 0\}$. Then exactly one of the following holds. (i) $\lambda(D) = 0$: the two candidates agree P-almost surely, deliver the same certified value on almost every instance, and any disagreement is confined to a Lebesgue-null set. (ii) $\lambda(D) > 0$: then $p := P(D) \ge \rho\, \lambda(D) > 0$, and for i.i.d. draws $\theta_1, \dots, \theta_m \sim P$,*

$$\Pr\big[\Delta(\theta_j) = 0 \text{ for all } j \le m\big] = (1-p)^m \le e^{-pm}.$$

Moreover, with an agreement tolerance $\delta > 0$ the same bound holds with D replaced by $D_\delta = \{\theta : |\Delta(\theta)| > \delta\}$ whenever $\lambda(D_\delta) > 0$.

Proof. Apply Lemma 2 to both candidates and take a common refinement. The restriction of Δ to each full-dimensional cell C_k is then a polynomial q_k. If every q_k is identically zero, Δ vanishes outside the null boundary set, and case (i) holds. Otherwise, some q_k is a nonzero polynomial on the full-dimensional cell C_k. The zero set of a nonzero polynomial within C_k has Lebesgue measure zero. Therefore, $\lambda(D \cap C_k) = \lambda(C_k) > 0$, which establishes the premise of case (ii). Independence gives the probability bound directly. By (A1), each draw belongs to D with probability $P(D) \ge \rho\, \lambda(D)$, so all m draws miss D with probability $(1-p)^m$. The tolerance result follows by applying the same argument to the open set $\{|q_k| > \delta\} \cap C_k$. □

Remarks. (1) The dichotomy defines the certification semantics. Candidates that differ only on a null set are behaviorally indistinguishable and certify the same value almost surely. Thus, equivalence under the gate is almost-everywhere equivalence. (2) The base instance θ_0 is not sampled. Instance 0 is mandatory and evaluated deterministically, so a disagreement at the stated instance is detected with probability one, independently of the bound. (3) The detection power is $1-(1-p)^m \geq 1-e^{-pm}$. Increasing m through escalation increases this quantity. (4) If perturbations enter the constraint matrix, the cell-wise values become rational functions. The zero-set argument still applies to nonzero real-analytic functions. The dichotomy is therefore not restricted to affine perturbations, although our analysis does not rely on this extension.

A.2 POLICY-MATCHED CALIBRATION

Setup. A *policy* π is a measurable map from the complete record ω of a problem to a final admission score. The record includes the text, panel randomness, and instance randomness, all of which are internal to that problem. We write the score as $S = g_\pi(\omega) \in \mathbb{R}$. An escalation ladder is a policy in which borderline intermediate scores trigger additional instances or candidates before the final score is emitted, with a cap of K rounds. Calibration draws $\omega_1, \ldots, \omega_n$ i.i.d. from a by-construction law Q that includes *null* records observed with certainty. These null records are accepted certificates for solver-verified calibration problems whose certified values contradict the verified answers. Deployment evaluates test problems whose null records are exchangeable with the calibration nulls. The gate accepts a test problem when the conformal p-value computed from the calibration scores passes the selection procedure at level α (Jin & Candès, 2023).

Proof of Proposition 2. (i) is the defining exactness of the Clopper–Pearson interval for a binomial proportion: for fixed τ the selected calibration certificates are, under the sampling model, independent Bernoulli trials with parameter p_τ, and U_τ is constructed so that $\Pr(p_\tau > U_\tau) \leq \delta$ with no asymptotic step. (ii) The selection τ^* is measurable with respect to the calibration data, so the event $\{p_{\tau^*} > 2\alpha\}$ is contained in $\bigcup_{\tau \in T}\{U_\tau \geq 2\alpha \text{ selected while } p_\tau > 2\alpha\} \subseteq \bigcup_{\tau \in T}\{p_\tau > U_\tau\}$, whose probability is at most $|T|\delta$ by (i) and the union bound. The factor $|T|$ is the entire price of the data-dependent selection; the pilot grid has four points. (iii) Under clause (ii) of Assumption 1, deployment certificates at score $\geq \tau^*$ and calibration certificates at the same score are exchangeable with respect to the true/false label, so their false rates share the parameter bounded in (ii); clause (i) is the mechanism premise under which the extracted specification, and hence the score, is a function of the intended instance. (iv) With a growing null pool, the conformal p-value of a test certificate is uniformly valid under exchangeability alone, and Benjamini–Hochberg over these p-values controls the realized FDR at level α in finite samples (Jin & Candès, 2023); the grid certificate is the small-null surrogate the pilot deploys because the p-value floor $1/(n_{\text{null}}+1)$ with ten nulls cannot clear $\alpha = 0.05$. The policy-matching requirement of Lemma 3 below applies to both instruments; the pilot's grid certificate inherits it exactly as the at-scale conformal form does. □

Lemma 3 (Policy-matched calibration, formal)**.** *(i) If calibration and deployment apply the* same *policy π, the calibration scores $g_\pi(\omega_1), \ldots, g_\pi(\omega_n)$ and each null test score $g_\pi(\omega')$ are i.i.d., the conformal p-values are valid, and the selection procedure controls FDR at level α. (ii) If calibration uses a single-pass policy π_0 while deployment uses a ladder $\pi \neq \pi_0$, the guarantee can fail by an arbitrarily large factor: for every K there exist score laws and ladders for which the realized null-acceptance rate is $1-(1-\alpha)^K$ against a nominal α.*

Proof. (i) The function g_π is a fixed measurable map applied to i.i.d. records. The calibration-null and test-null scores are therefore exchangeable. The validity of the conformal p-values and FDR control of the selection procedure follow from the standard split-conformal argument (Jin & Candès, 2023). Two conditions are essential: π uses only the record of the current problem, with no cross-problem adaptivity, and the same map is applied during calibration and testing. (ii) Take null intermediate scores i.i.d. $\text{Unif}[0,1]$ across ladder rounds, let τ be the $(1-\alpha)$-quantile calibrated under π_0 (one round), and let the deployed ladder re-score any below-threshold problem up to K times, accepting on the first exceedance. A null problem is then accepted with probability $1-(1-\alpha)^K$, which approaches $K\alpha$ for small α and 1 as K increases, while the nominal budget remains α. Generating calibration trajectories under the deployed ladder and recalibrating τ_π restores case (i) because the ladder is again a fixed map applied identically during calibration and deployment. □

Remark. The counterexample reflects a common implementation pattern. Retry-until-agreement escalation takes a maximum over dependent scores near the threshold. Operationally, the same ladder used in deployment must also generate the calibration trajectories.

B PROTOCOL REGISTRATION AND SCORING

Preregistered decision criteria. The following criteria were fixed before collecting admission or evaluation data, and we report the outcomes without modification. K1: if the best label-free arm achieves less than 70% of the macro accuracy of the ground-truth arm, we withdraw the mechanism claim and reposition the work as selective prediction. K2: if the gate does not outperform majority vote downstream, we withdraw the consensus-mechanism claim and retain only the efficiency contribution. K3: calibration fails if the realized false-discovery rate exceeds 5% in point estimate or 10% at the 95% upper bound. One rerun with a stricter threshold is allowed and must be reported. K4: a nearly uniform family-by-error matrix weakens the decorrelation argument and must be reported as such. Sections 4.2 and 4.3 report K1–K3. Section 4.2 reports K4, and Table 3 provides the full matrix extracted from the released certification verdicts.

Certification pseudocode. Algorithm 1 specifies the complete certification procedure for one problem. Section 3 describes each step.

Algorithm 1 ADMITOR certification of one problem P

extract base-anchored parameter spec θ_0 from P; validate and freeze structural sizes
draw perturbed instances $\theta_1, \ldots, \theta_m$ around θ_0 (seeded); instance 0 is P itself
generate candidate programs $C_1, \ldots, C_k$ from distinct model families and strategies; probe each on θ_0, one repair round on failure
for each candidate C_i and instance θ_j **do**
    solve; record objective $V_i(\theta_j)$ and solver status
end for
keep instances informative for at least three candidates; form agreement graph on value-function equality across informative instances
find maximal cross-family clique Q; score $s =$ (families in Q, $|Q|$, informative count)
if $s \geq \tau$ for calibrated threshold τ **then**
    ACCEPT: certify value $V_Q(\theta_0)$, admit trajectory with certificate
else if candidates disagree with diagnosable structure **then**
    ABSTAIN with instance-level diagnosis
else
    UNINFORMATIVE: emit nothing
end if

Scoring rule. One released scorer evaluates every benchmark and experimental arm. A prediction is correct if it matches the published label within relative tolerance 10^{-4} or if rounding the prediction to the number of decimal places printed in the label reproduces that label exactly. Labels are frequently rounded. For example, an exact solver optimum of 10.3333 is correct when the printed label is 10.33. Predictions that cannot be parsed count as errors. We also release stricter (10^{-6}) and looser (10^{-2}) scoring tiers for sensitivity analysis.

Candidate-generation rules. We developed the panel prompt once on a five-problem development split and fixed it before data collection. The Pyomo model object is named `model` and is never reused as a loop variable. The prompt avoids reserved component names and blanket exception handling. A canonical HiGHS solve-and-extract template defines success by the availability of an objective value rather than by a status-enum comparison. Extraction returns base-anchored parameter ranges and never perturbs structural size parameters. If a candidate fails the base-instance probe, it receives one repair round with the error context included. The following two prompts are complete. Line breaks are adjusted for layout, and $\{\ldots\}$ denotes fill-in fields. Instructions for the solver stack are inserted for each family. The released repository contains the complete scripts.

Candidate-generation prompt (verbatim).

```
You are an expert in optimization modeling. Write a COMPLETE Python
script that defines a function solve(params) which builds and solves
the optimization problem below.

STRICT RULES:
1. Use {stack}.
2. Every numeric quantity listed in PARAMS must be read from the
   `params` dict (params["<name>"]); do NOT hardcode their values.
   Structural constants (set sizes, index structure) may be inline.
3. solve(params) must return a dict:
   {"objective": <float>, "status": "optimal"} on success;
   {"objective": None, "status": "<short reason>"} if infeasible/failed.
4. No printing, no file I/O, no network. Output ONLY one python code
   block.
5. Name the model object `model`; NEVER reuse `model` or `m` as a loop
   index or comprehension variable. Avoid Pyomo reserved attribute
   names for components (e.g. activate, deactivate, name, clone,
   index, display).
6. Do not wrap the whole body in a blanket try/except; let unexpected
   exceptions propagate (the harness captures them).
[structured strategy only]
7. Before the code, include as a Python comment a compact IR in JSON
   (sets, params, decision variables with types, constraints one line
   each, objective), then implement EXACTLY that IR.

PARAMS (name: meaning):
{param_desc}

PROBLEM:
{question}
```

Extraction prompt (verbatim).

```
You are an optimization-structure extractor. From the problem below,
list the NUMERIC PARAMETERS (costs, prices, capacities, demands,
budgets, coefficients) together with their ACTUAL VALUES as given in
the problem data.

For each parameter also propose a perturbation domain for sensitivity
testing, anchored at the true values: keep it physically meaningful
but STRESS-TEST constraint activation (allow sign changes only where
economically plausible, e.g. a net profit that may turn negative;
widen enough that different constraints become binding). The domain
must remain feasible-plausible: do not propose ranges that contradict
the parameter's role (e.g. capacities near zero while demands stay
large).

Index-set sizes and cardinalities (number of employees, projects,
cars, ...) must NOT be perturbed: either OMIT them from params
entirely, or freeze them with a degenerate abs range (lo == hi ==
base). Changing a size without resizing every dependent array
produces structurally invalid instances.

Output ONLY a JSON object, no prose, exactly this schema:
{
  "params": {
    "<name>": {
      "meaning": "<short>",
      "base": <number or list, the TRUE values from the problem>,
      "perturb": {"mode": "rel", "r": <0..1>}
                 or {"mode": "abs", "lo": <num>, "hi": <num>,
                     "integer": <bool>}
```

```
      (in "abs" mode, lo and hi are ABSOLUTE values on the same scale
       as base, NOT offsets or deltas around it)
    }
  },
  "objective_sense": "max" or "min"
}

PROBLEM:
{question}
```

A certified skill (excerpt). The following file is one of the 101 items in the gate-built library. It was distilled from admitted trajectories by the unmodified procedure of the host, with no manual editing. Workflow 2 and some implementation details are omitted for space, and typography is adjusted slightly. The complete file is included in the released library.

```
name: Bin Packing with Resource Activation
description: Model and solve resource minimization problems with
  capacity constraints using binary assignment and activation
  variables, producing exact or feasible solutions with clear
  verification steps.

# Workflow 1 (CP-SAT with Explicit Linkage)

## Modeling stage

### Strategy Overview
This workflow uses Google's OR-Tools CP-SAT solver, designed for
discrete optimization with Boolean logic. It models the problem with
separate binary variables for assignment and resource usage, linking
them via simple linear constraints.

### Step 1 - Define Core Variables
- Binary assignment variables x[i][j] for each item i and resource j.
- Binary usage variables y[j] for each resource j to track activation.

### Step 2 - Enforce Assignment and Capacity
- Exclusive assignment for each item i:
  sum(x[i][j] for j in resources) == 1.
- Capacity for each resource j:
  sum(weight[i] * x[i][j] for i in items) <= capacity.

### Step 3 - Link Assignment to Usage
- For each i, j add the implication x[i][j] <= y[j], so a resource is
  marked used if any item is assigned to it.

### Step 4 - Formulate Objective
- Minimize the number of used resources:
  minimize sum(y[j] for j in resources).

### Common Pitfalls
- Forgetting to link assignment to usage, which lets unused resources
  escape the objective.
- Setting an insufficient number of resources, causing infeasibility;
  initialize with a safe upper bound like the number of items.

## Solving stage

### Step 1 - Configure Solver
- max_time_in_seconds, num_search_workers, random_seed; for exact
  solutions set relative_gap_limit = 0.0.

### Step 2 - Solve and Check Status
- Accept only OPTIMAL or FEASIBLE; on infeasibility check the lower
```

Table 3: The preregistered K4 check: candidate arm by terminal outcome, mined from the 298 stored certification runs (zero additional model calls). The profiles differ significantly ($\chi^2 = 30.96$, $\mathrm{df} = 8$, $p = 1.4 \times 10^{-4}$).

Candidate arm	Clique	Value excl.	Infeasible	Partial crash	All crash
DeepSeek + Pyomo (direct)	164	17	71	5	41
GPT + Pyomo (structured)	154	9	75	9	51
Claude + Gurobi (direct)	172	30	73	1	22

Table 4: Human-audited attribution of all 22 disagreements. No case is a gate model defect or a legitimate alternative interpretation. The dominant category (d) is a property of the benchmark, not of a solver.

Audited root cause	Cases
(a) gate defect: text states a constraint the clique drops	0
(c) legitimate alternative reading of ambiguous text	0
(b) label error (one confirmed by exact re-derivation)	2
(d) decisive data absent from the printed text	15
(d) parameters printed at truncated precision	5

```
  bound ceil(total_weight / capacity).

### Step 3 - Extract and Verify Solution
- Read the objective, per-resource usage, and per-item assignment;
  recompute per-resource total weight against capacity.

### Common Pitfalls
- Not checking both OPTIMAL and FEASIBLE statuses.
- Failing to provide a time limit for large instances.
```

B.1 THE DATA-LAYER STAGE

The stage runs before Step 1 whenever $E \geq 2$ extractor families are configured. Each family independently maps the text x to a base specification: a key set of named parameters with shapes, the stated values θ_0, and a perturbation domain per key. Panel entry requires (i) identical key sets and shapes, (ii) elementwise agreement of the stated values within the scoring tolerance of Appendix B, and (iii) domain compatibility after the guardrails of Step 2. On agreement, the panel proceeds on the agreed specification exactly as in the pilot. On disagreement, the ticket is routed to ESCALATE with a data-layer diagnosis naming the disagreeing keys; no candidate is generated, so the stage costs $E - 1$ extraction calls on agreeing tickets and saves the full panel budget on disagreeing ones. The stage targets precisely the class-(d) mechanism of Section 4.3: a single extractor that fabricates omitted data produces a specification no independent extractor is likely to reproduce elementwise, whereas a faithfully encoded text pins the same specification for every family. The label-free bound screen of Section 4.3, built from the sign of the optimum and cost lower bounds implied by total demand, is the cheapest instantiation of the same principle and already flags nine of the twenty-two audited disagreements post hoc. The pilot runs $E = 1$; measuring the stage's operating point, its escalation rate on faithful texts against its detection rate on class-(d) texts, is the designated first experiment of the sequel.

C HOST REPRODUCTION DETAILS AND SUPPLEMENTARY TABLES

Case study: a label convicts the innocent. One instance in a widely used cleaned benchmark carries a published label of 200 although its optimum verifies by hand as 250 (full derivation in the release). This case touches every stage of our evaluation. In reproduction (Section 4.1) the standard host returns 250 and is scored as incorrect. At admission, the judges diverge exactly as designed. the ground-truth judge compares against the published answer and rejects the correct trajectories,

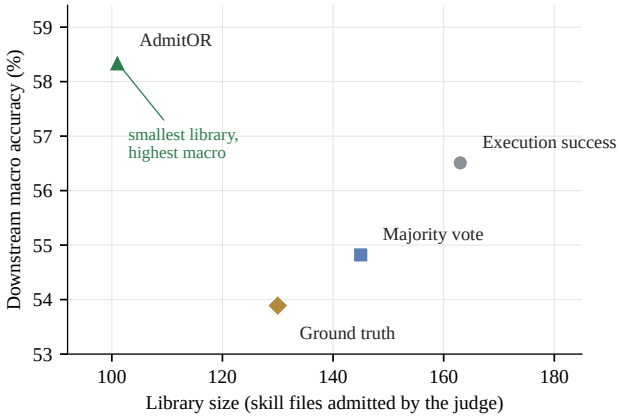

Figure 4: Library size and downstream macro accuracy for the four judges. Less selective judges admit more items and build larger libraries. The certified library is the smallest of the four and scores highest. Marker positions are the measured values of Table 2 and the library file counts of Section 4.2.

while the three families in ADMITOR agree on the sampled value-function trace and certify 250. The behavioral evidence therefore recovers a correct model that the erroneous label discards. At evaluation all four arms return 250 and are penalized, because the preregistered protocol retains the published labels. Label error can therefore move a small benchmark materially. The behavior of the gate follows from the design rather than from a favorable scoring choice, since it never reads the label.

The calibration replay supplies a complementary example. For one audited instance the certified and reference models differ by an entire constraint family, yet their objective values differ by 0.023%, and an exact solution of the fully printed instance reproduces the certified value. The reference therefore errs in this case. Similar objective values do not establish model equivalence, just as different values do not by themselves identify which model is wrong. The audit trail resolves this distinction.

Table 5 jointly summarizes the calibration and replay results from Section 4.3.

Table 5: Summary of E3. Left: self-labeling calibration on solver-verified instances. Right: replay of the calibrated rule on the wild-stream certification verdicts from Section 4.2, evaluated out of sample against the sealed vault with the uniform scorer (the stricter admission-time rule is shown in parentheses).

Calibration (NANO-CO)		**Replay (wild stream)**	
instances certified	150	accepts replayed	170
ACCEPT	103	admitted at τ	138
true cliques	93	coverage of accepts	81.2%
false cliques	10	disagreements	22 (26)
raw false rate	9.7%	realized FDR	15.9% (18.8%)
threshold τ	33.3	95% upper bound	22.0% (25.2%)
per-threshold δ	0.05	simultaneous level	$1 - \lvert T\rvert\delta = 80\%$

Table 6 reports the end-to-end reproduction of the host system with the released skill library. The repository does not provide scoring code, so the uniform round-aware scorer defined in Appendix B is applied to every benchmark and every downstream judge arm.

Annotations. **ComplexOR.** With $n = 18$, each item changes accuracy by 5.56 points. Therefore, falling within a ± 3-point band requires an exact match in the number of correct answers. The single difference is the instance discussed in Section 4.1. Its published label is 200, manual verification

Table 6: Host reproduction on five public benchmarks (accuracy, %). Published labels as-is; annotations below.

Benchmark	n	Reported	Ours	Δ
IndustryOR	100	36.00	36.00	0.00
Mamo.Complex	211	63.51	62.09	-1.42
OptiBench	605	77.02	75.87	-1.15
ComplexOR	18	72.22	66.67	-5.55
OptMATH-Bench	166	61.45	56.63	-4.82
Macro		62.04	59.45	-2.59

gives 250, and our pipeline returns 250. After correcting the label, the score becomes $13/18 = 72.22$, which exactly matches the reported value. Following the preregistered protocol, all main tables retain the published label. **OptMATH.** We rerun a cluster of 17 empty predictions in isolation with one worker. Only 3 change to correct answers, which is below the preregistered materiality threshold of 5. We therefore retain the original score. Solver contention explains at most 1.8 points of the deficit. The remaining difference reflects a capability gap in the reproduced pipeline on this benchmark. No central claim depends on the absolute OptMATH scores. An audit of the model identifier in each response confirms that every call used the fixed backbone. **IndustryOR.** Three labels contain the sentinel value -99999, which provides additional evidence of answer-key errors in widely used benchmarks. The release includes the complete run ledger, including call-level logs, segment-level resume records, and environment snapshots.