OPTIMAL FOR WHICH MEANING?

Anonymous authors
Paper under double-blind review

ABSTRACT

An optimization solver can solve a model correctly even when the model does not match what the user means. We study when different meanings change the recommended decision and when they can be left unresolved. We make three contributions. First, we separate three checks: whether the model expresses its stated assumptions, whether it is solved correctly, and whether its answer meets the user's requirements. Second, we give a rule for stopping clarification: the same action must satisfy every interpretation being considered and stay within an agreed loss limit. For problems with the same objective, this rule can reuse an optimization bound instead of solving every interpretation separately. Third, we test these ideas on existing model outputs and small controlled studies. In one production problem, an action that was optimal under the model's assumption earned 1,000 less under the cost rule confirmed by our collaborator. In six test cases, reusing bounds reduced verification solves from fourteen to six with the same fifteen valid stopping decisions. Giving the language model mathematical evidence helped one earlier task but did not consistently improve later corrections. Our results show why evaluation should check assumptions and decisions, not only the final objective value. Guarantees remain limited to the interpretations that were checked.

1 INTRODUCTION

A solver reports `OPTIMAL` when it has solved the model it was given. This does not mean the model answers the user's question. In an IndustryOR production problem, GPT assumes that a factory can avoid an equipment fee by not using that equipment. It solves this model correctly. Our collaborator instead confirms that all equipment fees must be paid. Under that rule, the proposed production plan earns 1,000 less than the best plan. The arithmetic is correct, but the assumption changes the decision.

Different models do not always require different decisions. Two readings of a sentence may allow the same plan to meet the user's needs. In that case, choosing between the readings may be unnecessary for this plan. On the other hand, two models may produce the same objective value while allowing different actions. Neither matching a number nor finding a difference between formulas is enough to decide whether an answer is useful.

This matters for how optimization assistants are evaluated. A scoring rule may respond to a change in mathematical notation even when the meaning is unchanged. An answer may match the reference value but make an incorrect claim about another reading. A system may also ask about details that do not affect the plan, or stop before resolving a difference that does. We provide concrete examples of these problems. We do not estimate how often they occur across all optimization requests.

Our question is: *which unclear requirements must be resolved before a system can recommend a plan?* We answer it for a stated set of interpretations and a stated limit on acceptable loss. Satisficing means meeting that requirement rather than always finding an exact optimum. Regret measures how much a plan loses compared with the best plan under the same interpretation. Robust feasibility means that the plan meets the constraints under every interpretation being considered. Together, these ideas let us ask whether the plan is good enough without requiring a unique meaning for every sentence.

Non-binding constraints suggest a way to reduce the work. Such a constraint has unused slack at a particular solution. But this alone does not show that its meaning can be ignored. Another reading

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.

may tighten it, require integer variables, or change how profit is counted. We instead check two things: whether the plan is feasible under all retained readings and whether its loss is small enough. When the objective is unchanged, a shared optimization bound can establish the second condition without separately solving every model.

We contribute a set of definitions and checks, together with experiments showing why the distinctions matter. The definitions separate a model's assumptions, its mathematical solution, and the requirements of the user. The stopping rule states when unresolved meanings do not prevent a satisfactory plan. The experiments examine scoring differences, an actual loss caused by an assumption, and the number of solves needed for verification. We also test whether showing mathematical evidence helps a language model correct its answer. An earlier task improves, but later comparisons show no added correction benefit. Our claims concern these findings and the stated stopping condition, not a general improvement over existing clarification systems.

2 LITERATURE REVIEW

Meaning and evaluation. AMBROSIA studies multiple query interpretations and DUST examines underspecification (Saparina & Lapata, 2024; Wildenburg et al., 2024). CheckList and distilled SQL test suites motivate behavioral checks beyond accuracy (Ribeiro et al., 2020; Zhong et al., 2020). We examine optimization consequences: different meanings can change feasibility, accounting, or decisions. EquivaMap, ModelEquivBench, and FLARE verify formulation relations (Zhai et al., 2025; Zhu & Xu, 2026; Robbins et al., 2026); our additional target is whether a verified conditional model supports its recommendation across remaining readings. Mathematical equivalence and decision adequacy are distinct, complementary relations.

Clarification and what-if analysis. OR-Clarify and InterOPT already address missing requirements, silent assumptions, and selective stopping (Ge et al., 2026). WIQOR and EOR study formulation modifications and decision consequences (Parashar et al., 2025; Zhang et al., 2025). We ask whether one action is acceptable *before* a reading is selected. We give a mathematical rule and tests for this question, rather than claim selective clarification itself as new. No matched interactive comparison against these systems has been conducted.

Decision theory and screening. Aspiration-based acceptance, robust optimization, regret elicitation, and overlapping decision regions provide established foundations (Simon, 1955; Bertsimas et al., 2011; Boutilier et al., 2005; Javdani et al., 2014). Safe screening likewise uses bounds to avoid unnecessary work in other optimization settings (Fercoq et al., 2015). The common-bound rule below applies classical principles to a rule for recommending a plan under several meanings. Its empirical test concerns reusable verification, not a new general screening theorem.

Clear and relevant answers. Grice's cooperation framework distinguishes evidence, informativeness, relevance, and clarity (Grice, 1975); its use in human–AI dialogue is already studied (Miehling et al., 2024). We use it to motivate relevant answers that state their assumptions, not to infer mathematical correctness from fluent conversation. An unresolved request can receive a clearly qualified conditional answer; the distinction between source ambiguity and ambiguity in the answer is essential. Appendix I specifies the proposed communication checks, which have not been separately tested.

3 ASSUMPTIONS

The mathematical claims apply under the following explicit conditions. Failure to establish a condition yields an unresolved claim, not permission to assume it.

A1: Declared interpretations and coverage. A finite nonempty $\mathcal{H}$ records readings linked to source text and assumptions. Review determines linguistic support, independently of reference-answer agreement. A guarantee over $\mathcal{H}$ covers actual intent $h^\star$ only if $h^\star \in \mathcal{H}$; an LLM-generated list is not assumed complete.

A2: Comparable actions and models that express the chosen meaning. All readings use visible business actions $x \in X$, with explicit mappings for auxiliaries. Reading h supplies feasibility F_h and minimized loss ℓ_h, retaining units and accounting periods; profit is negated. Agreement with the chosen meaning is checked separately from solver optimality.

A3: Well-defined acceptance. For regret, F_h is nonempty and $v_h = \min_{x \in F_h} \ell_h(x)$ is finite and attained. Thresholds are fixed before evaluating an action. Analyst tolerances are not user preferences. Infeasibility, unboundedness, and non-attainment are reported separately.

A4: Sound checks. Stopping uses globally valid bounds and feasibility under every retained reading, directly or through a valid aggregate test. Numerical tolerances are reported; incomplete verification is not proof. Algebraic and numerical evidence remain distinct.

A5: Stable clarification. A truthful answer filters R to nonempty $R' \subseteq R$, retaining $h^\star$ if present. Monotonicity assumes an unchanged action space and acceptance requirements. New interpretations, changed objectives, or revised tolerances require rechecking. Reliable unrestricted answer parsing is not established by our experiments.

4 Model setup

4.1 Meanings and mathematical models

We distinguish four source conditions. *Semantic ambiguity* permits multiple readings of an expression, such as a limit on the total versus a separate limit on each type. *Underspecification* leaves a decision-relevant value or convention unstated. *Inconsistency* supplies requirements that cannot jointly hold. A *modeling error* violates the selected reading. These labels can co-occur, but a model's alternative assumption does not by itself prove that the text is ambiguous.

Write feasibility as an arithmetic predicate $\Phi_h(x)$. The question $\forall x\,[\Phi_a(x) \leftrightarrow \Phi_b(x)]$ concerns feasible-set equality. A point satisfying only one predicate is a feasibility witness. One-way implication identifies containment. This is a logical relation, not a scalar "FOL score," and it says nothing about objective equality. Integer domains, units, and quantifier ranges must be explicit.

For a compiled model with auxiliary variables z, constraints $C_h(x, z)$, and loss $c_h(x, z)$, we use the condition that the model preserves the chosen meaning

$$\{(x, u) : x \in F_h,\ \ell_h(x) \le u\} = \{(x, u) : \exists z\,[C_h(x, z),\ c_h(x, z) \le u]\}. \tag{1}$$

The left set lists each feasible business action and all cost limits it meets. The right set does the same for the generated model, allowing its extra variables. These sets are called epigraphs; projecting removes the extra variables. Equality means that both models allow the same visible actions at the same cost limits. It permits different numbers of auxiliary variables and rules out equating formulations merely because their optimal values match. It is stronger than sharing optimizers: rescaling an objective requires an explicit conversion. A cost limit must actually be reached, not only approached. This condition checks the translation from a chosen meaning to a model; it does not settle the translation from the description to a chosen meaning.

4.2 Satisficing and regret under semantic uncertainty

For a supplied cost limit (an aspiration level) b_h, define $A_h = \{x \in F_h : \ell_h(x) \le b_h\}$. This is satisficing: the action must meet a requirement, not necessarily optimize. A regret limit instead sets $b_h = v_h + \tau_h$ for a tolerable loss $\tau_h \ge 0$. With a common dimensionless tolerance ε and fixed scales $s_h > 0$, set $\tau_h = \varepsilon s_h$ and

$$r_h(x) = \frac{\ell_h(x) - v_h}{s_h}, \qquad \mathcal{S}_\varepsilon(R) = \bigcap_{h \in R} \{x \in F_h : r_h(x) \le \varepsilon\}. \tag{2}$$

An infeasible action is unacceptable regardless of its reported objective. The scales must not be chosen retrospectively to make a candidate pass. Absolute tolerances avoid arbitrary normalization when objectives can be zero or shifted by fixed costs.

Definition 1 (An acceptable action)**.** *We call an action x acceptable (or decision-adequate) for retained interpretations R at tolerance ε if $x \in \mathcal{S}_\varepsilon(R)$. An unresolved distinction is irrelevant to this delivery if the action passes under every reading in R. This does not require equal feasible sets, equal optimal values, or equal sets of optimal actions.*

Robust feasibility is membership in $\cap_{h\in R} F_h$. Minimizing worst-case loss, $\min_x \max_h \ell_h(x)$, differs from minimizing worst-case regret,

$$\rho(R) = \inf_{x\in\cap_{h\in R}F_h} \max_{h\in R} r_h(x), \tag{3}$$

with $\rho = +\infty$ if no common feasible action exists. Satisficing asks whether a threshold can be met; it need not solve either minimax problem. Here the uncertainty set consists of meanings, not a probability distribution over physical outcomes.

Proposition 1 (Threshold and clarification)**.** *If the infimum in* (3) *is attained, then $\mathcal{S}_\varepsilon(R) \neq \varnothing$ if and only if $\rho(R) \leq \varepsilon$. Under A5, $R' \subseteq R$ implies $\mathcal{S}_\varepsilon(R) \subseteq \mathcal{S}_\varepsilon(R')$ and $\rho(R') \leq \rho(R)$.*

Without attainment, $\rho < \varepsilon$ is sufficient, whereas $\rho = \varepsilon$ need not be. Failure of one candidate is not evidence that the entire intersection is empty. A delivery may thus be verified, refuted for the proposed action, or unresolved. Only a global argument can rule out every common acceptable action.

4.3 What non-binding constraints do and do not justify

For $a_i^\top x \leq b_i$, non-binding at x means $b_i - a_i^\top x > 0$. Redundancy means removal leaves the feasible set unchanged. A difference is irrelevant to the current decision when the plan still meets the requirements after that change. These are different properties. For example, maximizing integer x subject to $0 \leq x \leq 2$ and $2x \leq 3$ gives $x = 1$, where the latter constraint has slack one. Removing it permits $x = 2$. Nominal slack is therefore insufficient even to justify deletion in an integer program.

A useful restricted setting keeps a common objective f and a relaxation F_0 with $F_h \subseteq F_0$ for all $h \in R$. This includes a confirmed core of requirements plus alternative additional restrictions. It also allows domain relaxations, provided the inclusion is valid.

Proposition 2 (A common bound for delivery)**.** *Let $L \leq \inf_{x\in F_0} f(x)$ and $F_h \subseteq F_0$ for all $h \in R$. If $\bar{x} \in \cap_h F_h$, then*

$$0 \leq f(\bar{x}) - v_h \leq f(\bar{x}) - L \quad \textit{for every } h \in R. \tag{4}$$

Consequently, $f(\bar{x}) - L \leq \min_h \tau_h$ is enough for the action to meet all loss limits. If $\bar{x}$ minimizes f on F_0 and remains feasible in every interpretation, it is optimal in each one.

The proof is the ordering $L \leq \inf_{F_0} f \leq v_h \leq f(\bar{x})$. The saving is reuse of one bound, not elimination of the semantic coverage requirement or of all feasibility checks. A loose bound can fail to certify an adequate action. Different objectives require separate valid bounds or an explicitly justified transformation.

For continuous linear programs, complementary slackness explains one route to bound reuse: $\lambda_i(b_i - a_i^\top x) = 0$. A changed right-hand side on a row with zero multiplier can leave a particular dual bound unchanged, provided the dual solution remains valid and the other problem data support its reuse. The primal action must still pass the changed constraints. A zero multiplier is not permission to change arbitrary coefficients, objectives, or integer domains. Checking "most" binding constraints is not a guarantee.

4.4 A worked semantic example and a clear-language control

In OptiBench O012, let integer $r, p \geq 0$ denote regular and premium handbags. Maximize $g = 30r + 180p$ subject to $200r + 447p \leq 250000$. The disputed cap gives

$$A:\ r + p \leq 475, \qquad B:\ r \leq 475,\ p \leq 475. \tag{5}$$

Two OR reviewers, as reported by the collaborator, preferred different readings but accepted the alternative as reasonable. Under A the optimum is $(0, 475)$ with profit 85,500; under B it is $(188, 475)$

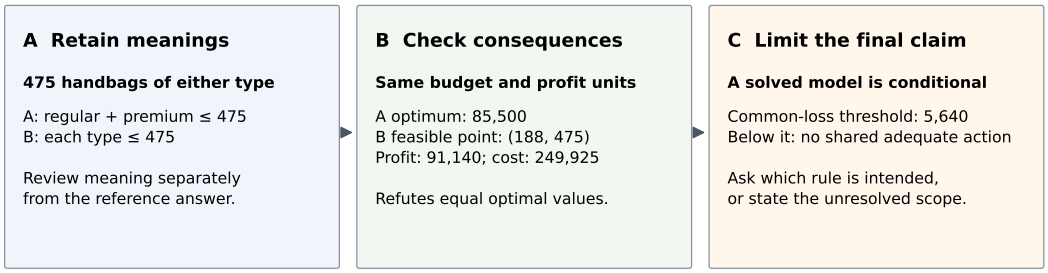


Figure 1: The running example separates retained meanings, a mathematical counterexample, and the scope of delivery. A better feasible point refutes a claim that both readings have the same optimum without proving the alternative optimum. A common regret threshold is a mathematical result, not an elicited user preference.

with profit 91,140. The latter costs 249,925 and violates A. Since $F_A \subset F_B$, the smallest common absolute regret is 5,640, attained at $(0, 475)$. Below that threshold no common adequate action exists. No user authorized this loss; the threshold describes the decision consequence of unresolved scope.

MAMO EasyLP1 is an explicit control, confirmed clear by the collaborator. It minimizes $50X+30Y$ over nonnegative integers with $X - Y \geq 200$, $X + Y \leq 1000$, $X \leq 700$, and $Y \leq 500$. The lower bound $50X + 30Y = 50(X - Y) + 80Y \geq 10000$ is attained at $(200, 0)$. Removing the three upper bounds preserves optimality. Tightening $X \leq 700$ to $X \leq 200$ also preserves it, although the constraint is now binding; $X \leq 199$ makes the problem infeasible. These controlled changes illustrate the distinction between slack and decision effect. They are not claimed to be alternative natural readings of the original text.

A change in accounting can change the action. IndustryOR 23 supplies a complementary case in which the objectives differ (Huang et al., 2025a). With integer production $x \in \mathbb{Z}^3_+$, labor and material requirements are $3x_1 + 2x_2 + 6x_3 \leq 1500$ and $4x_1 + 3x_2 + 6x_3 \leq 1600$. Contribution margin is $g(x) = 60x_1+40x_2+100x_3$. Reading A charges equipment costs only for active products; B always charges all costs:

$$g_A(x) = g(x) - \sum_{i=1}^{3} c_i \mathbf{1}[x_i > 0], \quad g_B(x) = g(x) - 4500, \quad c = (2000, 1500, 1000). \tag{6}$$

The collaborator selected B; A is retained as the model's actual assumption and an alternative to compare before clarification, not a second confirmed intent. An original GPT-5.4-mini run chooses A and returns $(0, 0, 250)$, earning 24,000 under A but 20,500 under B. A clarified B run returns $(100, 0, 200)$ and the B optimum 21,500. Hence the original action's loss under the confirmed convention is 1,000, not the difference between the two reported objective values.

Before selecting a reading, the two-model minimax-regret threshold is 840, attained at $(0, 99, 217)$, with regrets 840 and 340 under A and B. This illustrates why a plan that works under both readings need not be optimal for either. After B is confirmed, retaining the compromise would unnecessarily lose 340. The shared-objective bound does not apply directly to these differing objectives; separate conditional evaluation is required. Appendix G gives exact enumeration and review limitations.

4.5 When to stop and when to ask

Given reviewed R and a set of acceptance requirements, construct and validate the conditional models. Propose an action and check its feasibility under R. If a common bound proves its loss acceptable, deliver the action with the set of readings and tolerance to which the result applies. Otherwise, use per-reading optimization or return an unresolved status. When a violation is found, a witness can ground a question about the relevant rule. A truthful answer filters R, after which the check is repeated. If the interpretation set itself is inadequately supported, ask about coverage rather than equate absence of a discovered counterexample with safety.

The experiments test bound reuse, conditional generation, and evidence-assisted review separately. Automatic coverage of R and reliable parsing of unrestricted user replies remain outside the demonstrated interface.

5 EXPERIMENTAL DESIGN

We report development studies with different units of analysis rather than pool them into a single accuracy. Orbench initially motivated the project; the auditable studies emphasized here use public OptiBench, LeanOpt, and MAMO sources (Yang et al., 2025; LEAN-LLM-OPT contributors, 2026; Huang et al., 2025b). An earlier Orbench answer-scope labeling pilot is retained as development evidence, not an independently reviewed test set. Repetitions and numeric variants do not count as independent problem families.

What each experiment tests. We test four specific claims: whether a score can distinguish mathematically equivalent representations (archived affine audit); whether a conditional optimum can be inadequate for a confirmed intent or accompanied by a false cross-reading claim (IndustryOR and O012); whether a shared bound reduces repeated optimization (cached audit); and whether supplying a counterexample improves repair beyond targeted review (paired ablations). These are separate studies, not components of a measured end-to-end system.

Checking scores for different model representations. We recover an earlier audit of archived ORLM outputs on NL4Opt (Huang et al., 2025a; Ramamonjison et al., 2022). Rational affine reconstructions and aligned visible variables support comparison of projected epigraphs. Unsupported bindings or syntax remain unresolved. Of 245 archived records, 244 match a reference, 212 support semantic comparison, and 204 support both this comparison and the released scorer. A pair is scorer-distinguished when $S(C, R) \neq S(R, R)$; this is not a binary error label. We recount the saved per-pair results; we do not rerun the model-equivalence checker. Human reconstruction sign-offs are incomplete, so the classifications are reported as audit outputs, not independently human-confirmed equivalence. Details and controlled transformations appear in Appendix F.

Human review and mathematical checks. Review records distinguish preferred reading, permitted alternatives, and unresolved intent. Two reviewers' O012 judgments were relayed through the collaborator; no raw independent annotation forms were collected. IndustryOR's unconditional fixed costs and L067's one-to-one assignment were confirmed by the collaborator. L082 and L084 remain uncertain. MAMO EasyLP1 was confirmed clear by one collaborator, while other queued MAMO cases remain unread. Solver checks and algebra establish consequences conditional on these readings, not linguistic acceptability.

Checking the loss caused by a model assumption. The IndustryOR original and clarified runs use the same tool-using GPT-5.4-mini setup, without reference answers. We cross-evaluate the returned actions under A/B, rather than subtract incomparable reported objectives. Each condition has one generation; the comparison establishes a concrete mechanism, not an average effect across new problems. Integer production is an analyst assumption, shared by these runs and by the robust calculation. It was not separately elicited.

Study 1: Cached verification of supplied models. Six cases supply fourteen models and fourteen anchor actions, each checked at two analyst-selected tolerances, giving twenty-eight settings. We compare cached full per-interpretation optimization, cached sequential verification with early exit, and a common-bound method. All receive the same candidates and acceptance thresholds. We record valid stops, invalid stops, unresolved safe cases, optimization solves, and feasibility checks. The relaxation is chosen by the analyst. Some domain variants contradict explicit text and are mathematical negative controls, not licensed semantic alternatives. Fourteen anchor-generation solves are common preprocessing and reported separately.

Study 2: Generation and clarification on O012. DeepSeek receives the original description or an explicit clarification of cap A/B, using either a direct rule or a response to a concrete sales example. Five conditions have two repetitions each. Four intervention conditions share the same A/B

alternatives; user responses are scripted. The model calls a Gurobi-backed tool, with at most two tool submissions and a 600-second limit per solve, and receives no reference answer. We independently check generated models and solutions. This tests response to supplied rules, not whether real users understand a witness better.

Study 3: Paired review of fixed answers. An earlier 45-call pilot tests scope labeling on fifteen initial Orbench answers (Appendix H). The subsequent O012 repair study uses one actual answer with a false invariance claim. Three arms receive the same initial answer: targeted self-review; explicit alternatives; alternatives plus a checked better feasible point. Each has four repetitions. The second uses eleven inputs: three constructed false claims, two correct controls, and all six real outputs from Study 4. Each input is reviewed once per arm. The latter experiment allows one call and 4,096 output tokens per arm, without a new solver tool. The main measure is the verdict on a mathematical claim chosen before review, not full-answer correctness or intent recovery. Truncated outputs remain failures to complete. Actual token consumption differs, and evidence-construction costs are not included in API charges.

Study 4: Subsequent original problems. Three LeanOpt families (assignment L067, routing L082, and scheduling L084) are each run twice using the same generation prompt and full source tables. All six outputs are retained. Independent enumeration or lower bounds supplement numerical optimization. Their descriptions had been screened before execution, so this is a subsequent development test, not an untouched confirmatory sample.

Study 5: Rescoring and non-binding controls. We rescore the same sixteen outputs from Studies 2 and 4. We separately record reference-objective agreement, conditional mathematical correctness, and the scope of claims about other interpretations. The eight original-description runs are separated from eight clarified interventions: the old reference is not gold for a deliberately changed description. Five local MAMO solves provide the clear-language non-binding control described above. No additional LLM calls are made for this audit.

Implementation and measures. Except for the IndustryOR GPT case, the recent generation and review studies use the recorded OpenRouter identifier `deepseek/deepseek-v4.1-flash`. We preserve prompts, completion status, tool results, and source identifiers. A reference match concerns the numerical objective only. Conditional correctness requires the selected model and solution to implement its stated assumptions. Decision adequacy requires the reviewed interpretation set and explicit acceptance requirements. Where intent or tolerances are unknown, the decision label remains unknown. API truncation, invalid tool arguments, wrong mathematics, and unsupported cross-interpretation claims are separate outcomes. Appendix C gives further details and costs.

6 FINDINGS

A matching score does not tell the whole story. The archived audit classifies 178 jointly supported pairs as equal epigraphs; twelve receive a scorer response different from the reference self-comparison. The other jointly supported pairs comprise seventeen classified as different and nine unresolved, all scorer-distinguished. The twelve cases document representation sensitivity of this scorer under the reconstruction, not twelve universally wrong benchmark labels. Conversely, the O012 generation below shows that matching a reference objective can coexist with an incorrect claim about another reading. Together these observations motivate evaluating what relation a score actually measures.

A wrong assumption can lead to a worse plan. The IndustryOR original action incurs loss 1,000 under the collaborator-confirmed convention; explicitly stating that convention yields zero loss in the clarified run. A solver-optimal action is therefore not sufficient evidence of intended optimality. The computed threshold 840 also demonstrates a distinct option before clarification: a common satisfactory compromise, if a user permits that loss. These are case-specific existence results, not estimates of model accuracy or user willingness to accept regret.

Reusing a bound reduces the number of solves in this test. Table 1 reports the cached audit. The common-bound method uses six verification solves rather than fourteen and reaches the same fifteen

Table 1: Cached audit: 28 anchor–tolerance settings from 6 cases. Optimization counts exclude the 14 shared anchor-generation solves. Safe unresolved means an adequate candidate was not accepted.

Method	Solves	Feas. checks	Valid stops	Invalid stops	Safe unresolved
Full per-reading	14	72	15	0	0
Sequential per-reading	14	66	15	0	0
Common bound	6	72	15	0	0

Table 2: Paired selected-claim review. Entries count correct verdict on the selected claims; truncations are listed separately. Eleven inputs comprise five controls and six reused real outputs, not eleven independent problem families.

Input group	Allocated per arm	Self-review	Alternatives	+ Evidence
Constructed false claims	3	3	3	3
Constructed correct claims	2	2	1	2
Actual conditional outputs	6	6	4	5
Total correct verdicts	11	11	8	10
Wrong completed verdicts	-	0	0	0
Truncated outputs	-	0	3	1

valid stops. Candidate construction requires fourteen additional solves shared by all methods. The result demonstrates reusable bounds in these supplied, favorable model families; it is not a runtime or end-to-end interaction comparison. The method still performs seventy-two feasibility checks and does not avoid enumerating interpretations in this audit.

A correct numerical solution can accompany an incorrect semantic claim. Both original O012 runs solve A correctly. One additionally claims that B has the same optimum. The feasible point $(188, 475)$ refutes that statement. Across all ten O012 conditions, nine runs finish with a correct conditional model and value; one direct-B run fails to produce a completed final answer after invalid tool submissions. The witness conditions finish four of four, and the direct-rule interventions finish three of four. This difference is an interface failure in a tiny pilot, not evidence that witnesses improve semantic understanding.

Mathematical evidence helps some tasks, but not all. In the earlier Orbench pilot, supplied facts yield correct conditional-scope labeling on all five poster-area answers, versus zero under either comparison arm. This is one family and a strict check of the two required output fields, with greater realized token use in the evidence arm. It is a local positive result, not a general correction advantage. Appendix H preserves the complete comparison and other families. In the fixed O012 repair study, each arm corrects all four repetitions. The stronger review study likewise supplies no correction advantage for evidence (Table 2). All arms reject the three constructed false claims. All twenty-nine completed verdict on the selected claims are correct, but B truncates three times and C once. Correct verdict on the selected claims do not imply error-free explanatory prose: one evidence-assisted review confuses a statement in the supplied evidence with a fault in the submitted answer. We therefore retain the null result rather than claim that counterexamples teach a generally better reasoning strategy.

Conditional optimality does not settle missing intent. The six subsequent LeanOpt runs all match independently checked conditional values: 9,000 for assignment, 199 for a tour of the three named customers, and 22.85 for whole-task scheduling under the stated processing-time convention. The routing table includes additional locations beyond those named in the text; the scheduling description leaves implementation conventions unstated. Neither issue licenses treating every imagined alternative as intended. Because the reviewer retained uncertainty for these families, six conditional successes are not six verified intent matches.

Check the answer against the description it received. Among eight original-description outputs, all eight match their reference objective and the selected conditional mathematics. One nevertheless

contains the false O012 cross-interpretation claim. Among eight clarified interventions, four match the old reference, three disagree with it while correctly implementing the newly specified B rule, and one is incomplete. The three disagreements are not false negatives of the original benchmark: its input was changed. This distinction prevents overstating the benchmark critique while showing why reference matching alone cannot evaluate the scope of an answer.

Slack alone does not tell us whether to stop. The five MAMO checks confirm the earlier algebra: deleting upper bounds or making the cap binding preserves the optimum, tightening beyond feasibility does not, and deleting the required lower difference changes it. This clear-language control isolates mathematical effect without alleging ambiguity in the source.

Implications for benchmark design. A useful record should include the source span, permitted readings and review status, models with action mappings, acceptance requirement, and checked consequences. When a change affects only reporting conventions, evaluation should not call it an action error. When multiple readings share an acceptable action, a clarification need not be mandatory for that delivery. When they do not, a minimal disambiguating sentence can specify the relevant counting domain, variable domain, or cost convention. Whether this procedure reduces authoring effort or improves model performance remains an experiment to conduct, not a finding of this study.

7 Conclusion

An optimization answer makes claims at several levels: it encodes a meaning, solves the resulting model, and recommends an action. The experiments show why these levels should be evaluated separately. Archived scorer responses can change for formulations classified as equivalent; an actual solver-optimal action can be suboptimal under a confirmed cost convention; and a correct conditional answer can assert a false result about another reading. These failures would be obscured by treating a single numerical match as the entire evaluation target.

Our central question is simple: can the same plan meet the user's requirements under every meaning still being considered? Answering it requires more than matching a reference number. We must check the assumptions, the model, the solution, and the allowed loss. A constraint that is loose in one model can still matter under another meaning. When the objective is shared, a valid bound can reduce the number of solves needed to check a plan.

The experiments support specific conclusions. A wrong cost assumption can produce a worse decision even when the model is solved correctly. Reusing bounds saved solves in our small audit. Supplying mathematical evidence helped one earlier task but did not consistently improve later corrections. We have not shown how often these problems occur in new datasets, that a language model can find every relevant meaning, or that our method outperforms existing clarification systems. Those claims require new problems, independent review, and user-defined loss limits.

REPRODUCIBILITY STATEMENT

The accompanying local package records aggregate evidence, experimental identifiers, prompts where available, source revisions, and proofs. Gurobi results are numerical unless supplemented by the stated enumeration or algebraic arguments. API identifiers and costs describe the recorded runs rather than a promise of reproducibility under future provider changes. The present manuscript is a consolidated research draft, not a completed independent benchmark release.

AI ASSISTANCE STATEMENT

AI tools were used for research ideation and writing polish, with additional assistance in code and analysis. The authors are responsible for checking the claims, results, and references. AI assistance does not replace independent expert review.

ETHICS AND DATA STATEMENT

Judgments attributed to reviewers were communicated through the research collaborator; no controlled participant study or satisfaction study was conducted. Reviewers' identities are not disclosed. Private Orbench data motivated earlier development but are not redistributed here. Public-source usage and license conditions must be checked before a dataset release. An action verified only over retained interpretations must not be presented as an unconditional guarantee about an unobserved user's requirements.

REFERENCES

Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011. doi: 10.1137/080734510. URL https://epubs.siam.org/doi/10.1137/080734510.

Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. Regret-based utility elicitation in constraint-based decision problems. In *Proceedings of IJCAI*, 2005. URL https://www.ijcai.org/Proceedings/05/Papers/0398.pdf.

Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Mind the duality gap: safer rules for the lasso. In *Proceedings of ICML*, volume 37 of *PMLR*, pp. 333–342, 2015. URL https://proceedings.mlr.press/v37/fercoq15.html.

Sihan Ge, Yichen Lin, Chenyu Zhou, Jianghao Lin, Tao Yao, and Dongdong Ge. Ask before you optimize: Dynamic pre-formulation clarification for interactive optimization. *arXiv preprint arXiv:2609.05258*, 2026. URL https://arxiv.org/abs/2609.05258.

H. Paul Grice. Logic and conversation. In Peter Cole and Jerry L. Morgan (eds.), *Syntax and Semantics, Volume 3: Speech Acts*, pp. 41–58. Academic Press, 1975. URL https://web.stanford.edu/class/psych205/papers/Grice-1975.pdf.

Chenyu Huang, Zhengyang Tang, Shixi Hu, Ruoqing Jiang, Xin Zheng, Dongdong Ge, Benyou Wang, and Zizhuo Wang. ORLM: A customizable framework in training large models for automated optimization modeling. *Operations Research*, 2025a. doi: 10.1287/opre.2024.1233. URL https://doi.org/10.1287/opre.2024.1233.

Xuhan Huang, Qingning Shen, Yan Hu, Anningzhe Gao, and Benyou Wang. LLMs for mathematical modeling: Towards bridging the gap between natural and mathematical languages. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 2678–2710, 2025b. doi: 10.18653/v1/2025.findings-naacl.146. URL https://aclanthology.org/2025.findings-naacl.146/.

Shervin Javdani, Yuxin Chen, Amin Karbasi, Andreas Krause, Drew Bagnell, and Siddhartha Srinivasa. Near optimal bayesian active learning for decision making. In *Proceedings of AISTATS*, volume 33 of *PMLR*, pp. 430–438, 2014. URL https://proceedings.mlr.press/v33/javdani14.html.

LEAN-LLM-OPT contributors. LEAN-LLM-OPT: Large-scale-or test data and code. Public GitHub repository, 2026. URL https://github.com/CoraLiang01/lean-llm-opt. Snapshot cdc5d5261e075317ab76e7cb1503992eeca7ce10, accessed September 26, 2026.

Erik Miehling, Manish Nagireddy, Prasanna Sattigeri, Elizabeth M. Daly, David Piorkowski, and John T. Richards. Language models in dialogue: Conversational maxims for human-ai interactions. *arXiv preprint arXiv:2403.15115*, 2024. URL https://arxiv.org/abs/2403.15115.

Aditya Parashar, Ari Kobren, and Natalia Kosilova. WIQOR: A dataset for what-if analysis of operations research problems. Oracle Labs publication record and manuscript, 2025. URL https://labs.oracle.com/pls/apex/f?p=94065:10:18082377147334:11529. The author page lists ICLR 2025; the accessible PDF is a submission version.

Rindranirina Ramamonjison, Timothy Yu, Raymond Li, Haley Li, Giuseppe Carenini, Bissan Ghaddar, Shiqi He, Mahdi Mostajabdaveh, Amin Banitalebi-Dehkordi, Zirui Zhou, and Yong Zhang. NL4Opt competition: Formulating optimization problems based on their natural language descriptions. In *Proceedings of the NeurIPS 2022 Competitions Track*, volume 220 of *Proceedings of Machine Learning Research*, pp. 189–203, 2022. URL https://proceedings.mlr.press/v220/ramamonjison23a.html.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of ACL*, pp. 4902–4912, 2020. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442/.

Henry Robbins, Connor Lawless, Madeleine Udell, and Ellen Vitercik. FLARE: Verifying MILP reformulations with LLM-based theorem proving. *arXiv preprint arXiv:2608.25220*, 2026. URL https://arxiv.org/abs/2608.25220.

Irina Saparina and Mirella Lapata. AMBROSIA: A benchmark for parsing ambiguous questions into database queries. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.19073. Datasets and Benchmarks Track.

Herbert A. Simon. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69 (1):99–118, 1955. doi: 10.2307/1884852. URL https://academic.oup.com/qje/article-abstract/69/1/99/1919737.

Frank Wildenburg, Michael Hanna, and Sandro Pezzelle. Do pre-trained language models detect and understand semantic underspecification? ask the DUST! In *Findings of ACL*, pp. 9598–9613, 2024. URL https://aclanthology.org/2024.findings-acl.572/.

Zhicheng Yang, Yiwei Wang, Yinya Huang, Zhijiang Guo, Wei Shi, Xiongwei Han, Liang Feng, Linqi Song, Xiaodan Liang, and Jing Tang. Optibench meets resocratic: Measure and improve llms for optimization modeling. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/3deb687c44d3687ace0729e5db3b4efd-Abstract-Conference.html.

Haotian Zhai, Connor Lawless, Ellen Vitercik, and Liu Leqi. EquivaMap: Leveraging LLMs for automatic equivalence checking of optimization formulations. *arXiv preprint arXiv:2502.14760*, 2025. URL https://arxiv.org/abs/2502.14760.

Yansen Zhang, Qingcan Kang, Wing Yin Yu, Hailei Gong, Xiaojin Fu, Xiongwei Han, Tao Zhong, and Chen Ma. Decision information meets large language models: The future of explainable operations research. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a48e5877c7bf86a513950ab23b360498-Abstract-Conference.html.

Ruiqi Zhong, Tao Yu, and Dan Klein. Semantic evaluation for text-to-SQL with distilled test suites. In *Proceedings of EMNLP*, pp. 396–411, 2020. doi: 10.18653/v1/2020.emnlp-main.29. URL https://aclanthology.org/2020.emnlp-main.29/.

Penglin Zhu and Jungang Xu. ModelEquivBench: Certifying multi-relational evaluation of LLM-generated optimization models. *arXiv preprint arXiv:2607.29431*, 2026. URL https://arxiv.org/abs/2607.29431.

A Definitions and boundary cases

Multiple optima. Different solver outputs need not identify conflicting decisions. If $F_a = F_b = [0, 1]$ and both losses are zero, every action is optimal in both models. Conversely, minimizing x and minimizing $1 - x$ on this interval both give value zero but disjoint optimal sets. Compare aligned action sets, not two solver-selected points or scalar objective values.

Objective shifts and units. Adding a constant preserves absolute regret: $(\ell_h(x) + c) - (v_h + c) = \ell_h(x) - v_h$. A relative scale based on $|v_h|$ is not invariant to this shift. We require scales specified independently and use absolute regret in the handbag example. A change of units requires a corresponding change of tolerance. Aspiration levels need not be defined relative to an optimum; a feasible action can be checked against a known aspiration without solving the optimization problem.

Non-attainment. For $F = (0, 1]$ and $\ell(x) = x$, the infimum is zero but no action attains it. A strict capacity phrase can cause a similar boundary issue. Such a case should be reported as an infimum, with a feasible approximation if appropriate, rather than silently replacing a strict inequality by a non-strict one. It lies outside A3.

Coverage. Two users can generate the same observed conversation yet have disjoint acceptable-action sets. An algorithm using only that conversation has the same information in both cases; no unconditional action is guaranteed acceptable to both. A discriminating question, a justified coverage assumption, or withholding unconditional delivery addresses the missing information. This elementary observation rules out a universal semantic guarantee from agreement among a finite set of candidate interpretations alone.

B Proofs and mathematical checks

B.1 Source-to-model transfer

Fix h and $t = v_h + \tau_h$. Taking the section at $u = t$ in Eq. (1) gives

$$\{x \in F_h : \ell_h(x) \le t\} = \{x : \exists z\, [C_h(x, z),\ c_h(x, z) \le t]\}.$$

Equality of the actual epigraphs therefore transfers threshold acceptance. Equality of closures can add an unattained boundary point and is insufficient. The current experiments do not establish this fidelity condition automatically for arbitrary language or nonlinear models.

B.2 Threshold and monotonicity

If $\bar{x} \in \mathcal{S}_\varepsilon(R)$, then $\max_{h \in R} r_h(\bar{x}) \le \varepsilon$, so $\rho(R) \le \varepsilon$. Conversely, a solution that reaches the minimum with value at most ε satisfies all acceptance conditions. For $R' \subseteq R$, removing conditions enlarges their intersection. For every action feasible under R, the maximum regret over R' is no larger. Minimizing over the enlarged feasible set proves $\rho(R') \le \rho(R)$. These are set-inclusion facts, not new active-learning bounds.

B.3 Common-bound rule

For every h, $F_h \subseteq F_0$ gives $\inf_{F_0} f \le v_h$, and candidate feasibility gives $v_h \le f(\bar{x})$. Combining with $L \le \inf_{F_0} f$ proves Eq. (4). For maximum profit, a common upper bound U gives $0 \le v_h - g(\bar{x}) \le U - g(\bar{x})$.

A failed sufficient test is inconclusive. Let $F_0 = [0, 10]$, $F_a = F_b = [5, 10]$, and $f(x) = x$. At $\bar{x} = 5$, the bound $L = 0$ gives gap five although true regret is zero. Tighter bounds or per-reading solves can resolve the failure. Thus failure to verify a candidate differs from absence of any common satisfactory action.

One optimization call can supply a shared bound for K interpretations, followed by K feasibility checks. This count excludes translation, candidate generation, and construction of the relaxation. A structured aggregate oracle might replace the individual checks, but is not implemented here. No sublinear guarantee in the number of meanings is claimed.

B.4 Complementary slackness: scope of reuse

For $\min c^\top x$ subject to $Ax \geq b$, $x \geq 0$, the dual maximizes $b^\top y$ subject to $A^\top y \leq c$, $y \geq 0$. If y is dual feasible and only b_i changes where $y_i = 0$, the bound $b^\top y$ is unchanged and dual feasibility is unaffected. A candidate that remains primal feasible and reaches the bound stays optimal; one within τ has regret at most τ.

This does not establish primal feasibility. Coefficient changes require renewed dual-feasibility checks; objective changes can invalidate the former dual. A continuous relaxation can bound an integer program, but its complementary slackness does not characterize integer optimality. These distinctions prevent extending a local slack observation into an unsupported semantic stopping claim.

B.5 Handbag arithmetic

For A, $g = 30r + 180p \leq 180(r + p) \leq 85500$, attained at $(0, 475)$. For B,

$$g = 0.15(200r + 447p) + 112.95p \leq 37500 + 112.95(475) = 91151.25.$$

Integer profits are multiples of 30, so $g \leq 91140$. The point $(188, 475)$ attains this bound at cost $37600 + 212325 = 249925$. Every common action belongs to A and hence loses at least 5,640 relative to B's optimum; $(0, 475)$ attains this worst-case regret and has zero regret under A.

The point $(300, 300)$ costs 194,100, satisfies B, and violates A. A question about its compliance separates these cap rules if other requirements are fixed. In contrast, $(188, 475)$ is a better feasible point that refutes the unchanged-optimum claim. Refutation does not itself require proving B-optimality.

B.6 MAMO arithmetic

For $X, Y \geq 0$, $X - Y \geq 200$ implies $50X + 30Y = 50(X - Y) + 80Y \geq 10000$. The integer point $(200, 0)$ attains the bound and satisfies $X + Y \leq T$, $X \leq U$, $Y \leq V$ whenever $T \geq 200$, $U \geq 200$, $V \geq 0$. If $U = 199$, infeasibility follows from $X \geq Y + 200 \geq 200$. The source states its limits explicitly; the parameter family is an analytical control, not a set of alleged natural readings.

C Experimental details and costs

C.1 Study ledger

Record	Unit	Role and limitations
v26	6 cases; 14 models; 28 settings	Cached mathematical audit; invalid domain variants remain negative controls.
v28	10 O012 generation tasks	Two original runs and eight scripted interventions.
v29	12 reviews of one v28 answer	Three arms, four repetitions; same false claim.
v31	6 outputs; 3 LeanOpt families	Conditional verification; two families' intent unresolved.
v33	33 reviews of 11 inputs	Five controls and all six v31 outputs; one call per arm/input.
v34	16 reused outputs; 5 solves	Rescoring v28/v31 and explicit MAMO controls.

Repeated O012 use and reuse of v31 outputs make these studies dependent. No confidence intervals treating repetitions as independent families are reported. The record diagnoses mechanisms rather than estimates population ambiguity frequency.

C.2 Cached audit

The six cases are L001, L050, L072, L081, O000, and O012. They concern domain/demand conventions, domain, cyclic staffing, fractional domain, price steps, and cap scope, respectively. The integer variant of L081 and continuous-price variant of O000 contradict explicit source conditions and remain negative controls after review.

Fourteen model optima supply anchors. Each uses $\tau = \eta \max(1, |v_{\text{anchor}}|)$ for $\eta \in \{0, 0.05\}$, fixed across its interpretation group. This is an experimental sensitivity parameter, not a user preference or universal regret normalization. All methods receive the same action and tolerance in a comparison. The analyst-supplied largest-set relaxation is favorable to reuse. Full and sequential methods use fourteen cached verification solves each; common bounds use six. These 34 calls plus fourteen shared anchor-generation calls yield 48 optimization calls. The saving does not automatically extend to non-nested model families.

C.3 Model generation

The frozen prompt requires the model to use `solve_optimization`; a structured formulation is compiled into Gurobi. It forbids invented solver results and arbitrary bounds, permits at most two submissions, and requires assumptions, selected reading, objective, action, and solver status. Missing essential data should produce a needs-information status. All runs use full available tables and no reference answers.

The O012 study uses 22 API calls, costing USD 0.013716254. Twelve tool submissions yield nine optimal solves and three schema errors. The nine completed models are independently compared after mapping variables to r, p: domains and bounds are checked, feasibility is compared on all $476^2 = 226576$ points, and objective expressions are checked. One witness-B output has correct model/value but reports manufacturing cost 250,000 instead of 249,925 in prose. Conditional model/value correctness is therefore not full-answer correctness.

The subsequent LeanOpt study uses 13 API calls, costing USD 0.020452224. Enumerating six assignments checks L067, and enumerating six customer orders checks L082. Extra table locations are not automatically mandatory visits. Both L084 runs use whole-task non-preemptive scheduling under the adopted instruction-rate convention. One initially incomplete tool submission is repaired; a final status string also differs from the requested schema despite valid mathematics.

C.4 Scheduling lower bound

Workloads total 136.7 billion instructions in increments of 0.1. Completion before 22.85 seconds would require loads below $1.33(22.85) = 30.3905$, $2(22.85) = 45.7$, and $2.66(22.85) = 60.781$. Discrete caps would then be 30.3, 45.6, and 60.7, totaling only 136.6. The returned assignment achieves loads 30.3, 45.7, and 60.7, and makespan 22.85. This proves conditional optimality, not that clock frequency alone specifies instruction throughput on real hardware.

C.5 Answer review

In v29 every arm receives the same original transcript and a targeted request to check its invariance claim. B explicitly states the alternatives; C also supplies a better feasible point and its scope. All arms retract the false statement four times. A/B/C output tokens total 11,013/8,313/5,396; input tokens total 6,208/6,432/6,740. Total API cost is USD 0.010749816. Actual token use is not equal.

In v33 the common prompt requests SUPPORTED/REFUTED/UNKNOWN for a selected claim, a reason, a corrected claim, delivery status, and evidence scope. Conditional support is explicitly separated from unstated user intent; a new solver run must not be claimed. Each arm gets one call, the same reasoning setting, and a 4,096-output-token cap. The three false controls concern O012 optimum invariance, O012 common feasibility, and an explicit price step. The two correct controls concern O012 A-optimality and integrality of a standard feasible bounded transportation LP with integer data and no additional restrictions. The transportation item is a mathematical control, not observed ambiguity.

A/B/C costs are USD 0.011952475/0.0246152912/0.0182842364, totaling 0.0548520026. Truncations occur for one correct-control B review and three L084 reviews (two B, one C). They remain in the denominator. All completed verdict on the selected claims are correct, but an evidence-assisted explanation misattributes evidence text to the submitted answer. Evidence construction and analyst time are not included in API cost.

Group	Count	Interpretation
Original descriptions	8	All reference objectives and checked conditional mathematics match; one false cross-reading claim.
Clarified, old-reference match	4	Explicit rule retains the old optimum.
Clarified, old-reference mismatch	3	Correct for newly specified B; not false negatives on unchanged inputs.
Clarified, incomplete	1	No final answer; not classified as semantic error.

C.6 Scoring audit

Unknown intent or tolerance remains unscored. A conditional answer is not an unconditional robust delivery unless it makes that claim. Neither reference disagreement nor a mathematical difference alone establishes wrong delivery.

D Source and review provenance

LeanOpt uses public revision `cdc5d5261e075317ab76e7cb1503992eeca7ce10`; MAMO uses `2c91069dee5c1e4db4b899eb176ec987d0251bb3`. Its downloaded LP files contain 652 easy and 211 complex records; the search inspected the first six easy and first two complex records, not all problems. Related marketing variants are one family. Only EasyLP1 has the clear-language review used here. EOR revision `2db57e87f8bdbcd4602c54d6e787669efb3b92e8` supports a review queue, not a performance result. WIQOR is related work rather than an evaluated dataset. Earlier Orbench, IndustryOR, and NL4Opt explorations informed development but are not additional held-out evidence in this draft.

O012 judgments were communicated by the collaborator from two OR reviewers, without independent forms or a blinding record. The collaborator confirms L067 and MAMO EasyLP1; L082 and L084 remain uncertain. An unread case is not labeled uncertain or ambiguous. No tolerance used here was elicited in a controlled participant study.

E Requirements for a confirmatory study

Freeze elicitation, prompts, scales, and stopping rules. Select new families before observing outcomes. Independently review preferred versus permissible readings and the corresponding models. Include clear text, ambiguity with and without a common acceptable action, and missing-data or non-attainment cases. Compare the same actions and readings using cached full optimization, sequential verification, nominal-slack skipping, and a common bound. Separately compare interactive methods with the same model, questions, user responses, and budget. Report wrong unconditional delivery, justified delivery, unresolved cases, unnecessary clarification when defined by stated requirements, and total cost including model translation and evidence generation.

For authoring, compare original text, minimal decision-relevant clarification, full-slot clarification, and a length-matched irrelevant addition. Score against each actual input rather than a stale reference. Test reductions in interpretation disputes and unacceptable actions. This proposed study is not a finding of the scripted O012 interventions.

F Affine representation audit

This analysis restores an earlier internal audit as a component of the present study; it does not cite an unpublished companion paper. It concerns archived ORLM/NL4Opt output pairs and a released formulation scorer, not the reference-objective metric used in the subsequent generation experiments. Static reconstruction maps supported rational affine constraints and visible variables into an action–cost representation. Unsupported syntax and bindings do not become negative equivalence judgments.

Of 245 raw archived records, 244 match a reference. Semantic reconstruction supports 212 comparisons and the scorer additionally supports 204. The row-level cross-tabulation is: The equal-

Scorer response	Equal epigraph	Different epigraph	Unresolved
Distinguished	12	17	9
Not distinguished	166	0	0

Table 3: Recorded classifications on 204 jointly supported pairs. Scorer distinction means $S(C, R) \neq S(R, R)$, not necessarily a binary incorrect label.

epigraph column contains 178 pairs. We recount the results in this ledger with assertions on the cross-tabulation; this is not a new semantic-oracle run. A separate reconstruction precheck records 42 automatic passes and ten cases needing human review; final human sign-offs are blank. Neither the 204 rows nor the 178 equal classifications should be described as a completed independent human audit.

The controlled ledger has 4,404 planned transformation positions, of which 3,685 are generated. Of these, 3,203 relations are decided and 482 remain unresolved. For decided positive row scalings, 1,038 of 1,041 alter the scorer response; for decided clause permutations, zero of 1,041 do. All 917 decided effective right-hand-side/coefficient changes are distinguished. These are checks performed after developing the method of a particular scorer and reconstruction, not a measure from a new test of benchmark error frequency. The shared sources and derived mutations prevent treating all ledger positions as independent examples.

This result supplies a concrete reason to separate encoding agreement from mathematical meaning. It does not establish the soundness of all source bindings, resolve semantic ambiguity in NL4Opt descriptions, or show that every discrepancy affects a decision. Its proper role is an auxiliary scoring diagnostic supporting the proposed evaluation distinctions.

G INDUSTRYOR DECISION-LOSS CHECK

The source provides product labor coefficients $(3, 2, 6)$, material coefficients $(4, 3, 6)$, selling prices $(120, 80, 180)$, variable costs $(60, 40, 80)$, and weekly equipment costs $(2000, 1500, 1000)$. Weekly capacities are 1,500 labor units and 1,600 material units. Integer production is a working assumption for the experiment. The collaborator confirms that all equipment costs must be paid regardless of production; this is one reviewer's specified intent, not independent evidence that the source has exactly one universally accepted reading.

The original GPT-5.4-mini result explicitly states the conditional-activation assumption, calls the solver, and reports $(0, 0, 250)$ with profit 24,000. Under unconditional costs the same action earns 20,500. The clarified run returns $(100, 0, 200)$ and 21,500. The added wording states that all three equipment costs must be paid every week, including when the corresponding product has zero production, and specifies integer quantities. The original run had already selected integer quantities, but the experiment is still a single-run mechanism check rather than a randomized wording-effect estimate.

Action	Profit A	Profit B	Regret A	Regret B
$(0, 0, 250)$	24,000	20,500	0	1,000
$(100, 0, 200)$	23,000	21,500	1,000	0
$(0, 99, 217)$	23,160	21,160	840	340

Table 4: All regrets evaluate an action and its comparator under the same cost convention.

A fresh integer enumeration included with this draft checks the two optima and minimum worst-case regret over every resource-feasible production triple. Finite limits follow from the nonnegative resource constraints: $x_1 \leq 400$, $x_2 \leq 533$, and $x_3 \leq 250$. The minimum worst-case regret is 840, so no common action meets an absolute tolerance of 839; $(0, 99, 217)$ meets 840. These computations depend on the integer-domain assumption. They do not authorize a user loss of 840 or justify retaining an obsolete interpretation after clarification.

The original activation formulation uses upper linking constraints rather than an exact iff for every auxiliary assignment. Positive fixed costs make the relevant optimal lifted choices agree with the intended activation semantics, but we do not claim pointwise equality of all extended assignments. Two recorded explanations also contain an incorrect floor calculation for a production bound; the actual model uses a valid loose bound and the resource constraint remains active. These prose defects are preserved in the underlying record rather than hidden by an optimality label.

H EARLIER PAIRED ANSWER-SCOPE LABELING EXPERIMENT

This study was included in the user-identified v23 manuscript and is retained to avoid selecting only the later null results. Fifteen completed initial answers from three Orbench development families received three paired repair conditions: targeted self-check, two readings, and the same readings plus analyst-checked facts. Five initial answers concern poster area, six rent increments, and four strict aquarium volume. Three incomplete initial answers were excluded before repair equally across arms. All arms use DeepSeek-V4.1-Flash, one additional call, the same reasoning setting and 4,096-output-token cap, without a new solver call. The 45 calls are not 45 independent problem families.

Outcome	Self-check	Readings	+ Facts
Poster: asks which area (of 5)	0	2	5
Poster: conditional scope and unknown original feasibility	0	0	5
Rent: no further question (of 6)	6	6	6
Volume: infimum and invalid boundary (of 4)	3	3	3
Completed outputs (of 15)	14	14	14
Output tokens	25,123	24,842	34,265

Table 5: Earlier answer-scope labeling results, distinct from the subsequent selected-claim verdict experiments.

The poster's stated area can be associated with the printed region or the entire poster. Some readings-only answers discard the latter because it makes paper area constant. The facts condition retains that possibility and labels the answer conditionally. Two readings-only answers ask the scope question yet fail the conjunction of required status fields; zero strict passes does not mean zero ambiguity recognition.

All arms truncate on the same volume trajectory; there is no retry. The evidence arm additionally asks one volume clarification without demonstrated necessity when a strict interpretation could be answered by stating non-attainment. Actual input/output tokens are not matched. API charges are USD 0.017617344, 0.0184577176, and 0.0260609288 for A/B/C, totaling 0.0621359904. Analyst-supplied evidence, one-family improvement, and different evaluation measures prevent pooling this outcome with subsequent review verdicts into a general efficacy estimate. Private source descriptions and raw provider transcripts are not included in the redistributed package.

I COMMUNICATION REQUIREMENTS AND MATHEMATICAL STOPPING

We distinguish a formal delivery gate from a communication check. The former establishes feasibility and loss within a declared interpretation set. The latter asks whether the response actually communicates that scope and addresses the request. The following is our proposed set of practical checks, not a theorem of conversational pragmatics:

- State only conclusions supported by the model and verification, with unresolved assumptions visible.
- Supply enough information to understand the action and its conditions without treating every conceivable detail as a required clarification.
- Explain how a question's answer could affect the requested action, loss, or scope of the claim. Other conversational purposes can still justify a question.
- Use explicit domains, units, counting ranges, and conditional language; a fluent but ambiguous qualification does not communicate the guarantee.

These requirements are not equivalent to saying the original request must become unambiguous. For example, a response can state that the proposed plan is feasible under both retained cap readings and within a declared loss bound without selecting one of them. Conversely, if the requester needs the exact intended optimum and the readings have no common optimal action, the unresolved choice matters to that task. Neither meeting the mathematical requirements nor these communication checks alone proves user satisfaction. We have not collected a new conversational-maxim annotation or evaluated a Grice-based prompting intervention.