

ModelEquivBench: Certifying Multi-Relational Evaluation of LLM-Generated Optimization Models

Penglin Zhu, Jungang Xu*

School of Computer Science and Technology, University of the Chinese Academy of Sciences
zhupenglin21@mails.ucas.ac.cn, xujg@ucas.ac.cn

Abstract

Large language models increasingly generate optimization models from natural language, but existing evaluation often reduces a generated model and its ground truth to a single *equivalent/not-equivalent* verdict or an execution-success rate—labels that are neither independently checkable nor faithful to the multiple distinct senses in which two formulations can agree. We present MODELQUIVBENCH, a certifying, multi-relational evaluation system that reports a per-pair *semantic profile* $E0$ – $E6$: model construction and exact ingestion ($E0$), verified representation alignment ($E1$), same-space and projected feasible-set relations ($E2$, $E3$), objective-order equivalence ($E4$), optimal-value equality ($E5$), and optimizer-set equivalence ($E6$). Each decided entry carries relation-appropriate, independently re-checkable evidence: replayable traces or explicit maps for $E0$ – $E1$, exact-rational certificates for positive $E2$ – $E6$ conclusions, and explicit witnesses for supported negatives. Incomplete mapping search, unsupported structure, and resource limits produce typed UNKNOWN or N/A outcomes rather than guesses, while unmet prerequisites are reported as ABSENT. Using MODELQUIVBENCH to evaluate three model snapshots—GPT-5.4, Claude Sonnet 4.6, and Qwen3.5-397B-A17B—on the same frozen cohort of 173 base problems (346 cells per model) under a no-repair protocol, the resulting profiles expose distinctions that coarse baselines do not represent: 49, 35, and 25 cells contain executable candidates that are nevertheless certified negative on at least one supported relation, and 25, 8, and 18 structural rejections occur on pairs for which $E2$ certifies mapped feasible-set equality under a verified map. The three model snapshots fail at different stages of the profile and therefore cannot be meaningfully reduced to a single accuracy score.

1 Introduction

Large language models (LLMs) are increasingly used to turn natural-language problem descriptions into runnable optimization models (Ramamonjison et al. 2022; AhmadiTeshnizi, Gao, and Udell 2024; Xiao et al. 2024; Huang et al. 2025). Assessing whether a generated model is *correct*, however, remains unsettled. Common signals include (i) *execution success*—the code

runs and a solver returns a number—and (ii) a single *equivalent/not-equivalent* verdict against a ground-truth model, produced by value comparison or structural graph matching (Wang et al. 2025; Zhai et al. 2025). Neither by itself answers all of the semantic questions relevant to formulation correctness. Execution success says nothing about whether the model *means* the right thing: a program can build, export, and solve a model that encodes the wrong feasible region or objective. A single global equivalence label, in turn, conflates several genuinely different questions—do the feasible sets coincide, do the objectives induce the same ordering, are the optimal values equal, are the optimizer sets in bijection?—and is typically returned without a proof that a third party could independently re-check.

We take a different stance. Given an LLM-generated (*candidate*) model and a *reference* model, we ask: *in which distinct semantic senses do they agree or disagree, and which of those conclusions can be independently certified?* Our answer is MODELQUIVBENCH, a certifying, multi-relational evaluation system that reports a per-pair *semantic profile* of seven dimensions, $E0$ – $E6$, summarized in Table 1. The dimensions range from model construction and exact ingestion ($E0$), through verified representation alignment ($E1$) and feasible-set relations in the aligned space ($E2$) and under an affine lift ($E3$), to objective order ($E4$), optimal value ($E5$), and optimizer sets ($E6$). Crucially, $E0$ – $E6$ form a *profile*, not a ladder: the indices order the dimensions’ *definedness* prerequisites, not their logical strength, and equal feasible sets do not imply equal objective order, value, or optimizers. There is no “deepest passing level” and no scalar collapse.

MODELQUIVBENCH is *certifying*: each decided entry carries relation-appropriate evidence that an independent checker can re-verify. $E0$ uses a replayable execution/ingestion trace, $E1$ an explicit admissible map, and positive $E2$ – $E6$ conclusions exact-rational certificates (Farkas, affine-lift, objective-identity, or primal–dual evidence). Supported negatives carry a failing trace or explicit witness. Incomplete search, unsupported structure, and resource limits yield typed UNKNOWN or N/A outcomes; failed prerequisites make later dimensions ABSENT. The bidirectional feasible-set containment engine

*Corresponding author.

we call *Certifying Mapped-Containment* (CMC) is the principal technical engine for $E2$ and the supported part of $E3$; it is a component inside MODELEQUIVBENCH, not a replacement for the $E0$ – $E6$ profile.

We instantiate MODELEQUIVBENCH on the supported envelope of linear and bounded-discrete models and evaluate three model snapshots—GPT-5.4, Claude Sonnet 4.6, and Qwen3.5-397B-A17B—on a frozen cohort of 173 base problems, each under paired Structured and Unstructured conditions (346 cells per model), with one generation per condition at temperature 0.0 and *no repair or resampling*. Their profiles differ sharply. GPT-5.4 produces ingestible candidates for 334/346 cells, compared with 156/346 for Claude Sonnet 4.6 and 196/346 for Qwen3.5-397B-A17B. Conditional $E1$ coverage is nevertheless similar: 277/334 (82.9%), 130/156 (83.3%), and 164/196 (83.7%). Qwen3.5-397B-A17B additionally records 17 provider/API errors before candidate creation; these are reported as $E0$ ABSENT, not model-quality failures. Exact certification reveals 49, 35, and 25 execution-success overestimations and 25, 8, and 18 structural rejections despite $E2$ -certified mapped feasible-set equality. Our contributions are: (1) MODELEQUIVBENCH, the $E0$ – $E6$ certifying multi-relational evaluator and its typed-abstention discipline; (2) a certifying implementation with independent re-verification of every decided fact; and (3) a three-model formal study showing distinctions that execution-only, value-only, and structural baselines do not represent. The contribution is an evaluator and certification system, not a new benchmark dataset: Bench4Opt supplies the experimental instances. Supplement Sections A–E give the full definitions, certificate soundness proofs, boundary cases, and explicit non-implication examples.

2 Related Work

2.1 LLM Optimization Modeling and Evaluation

LLM optimization modeling. A growing line of work prompts or fine-tunes LLMs to formulate optimization models from text, including the NL4Opt competition (Ramamonjison et al. 2022), agentic decomposition systems (AhmadiTeshnizi, Gao, and Udell 2024; Xiao et al. 2024), and trained modelers (Huang et al. 2025; Lu et al. 2025). These works focus on *generation*; we focus on *certifying evaluation* of generated models against a reference.

Evaluating generated mathematical programs. Evaluation has largely relied on execution success and optimal-value matching, or on a single equivalence verdict. EquivaMap uses an LLM to propose mappings between decision-variable spaces and then verifies feasibility and optimality preservation (Zhai et al. 2025); EquiBench studies LLMs’ program-equivalence reasoning (Wei et al. 2025). These approaches target an overall equivalence judgment, whereas MODELEQUIVBENCH reports separately certified relations and typed abstentions.

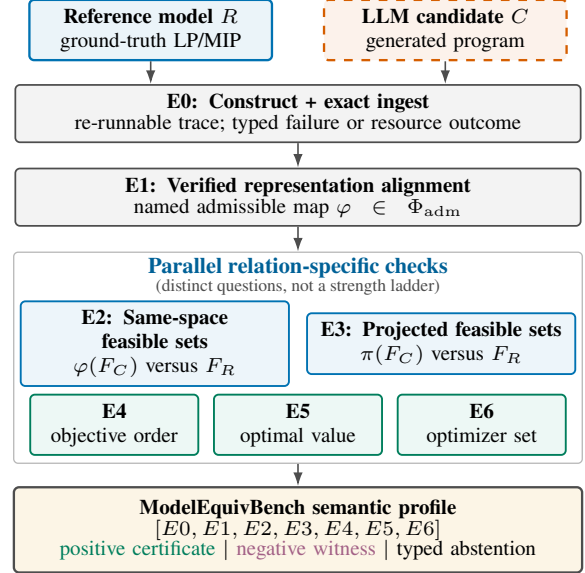


Figure 1: The MODELEQUIVBENCH certifying $E0$ – $E6$ workflow. $E0$ and $E1$ establish prerequisites; $E2$ – $E6$ ask parallel, relation-specific questions. Decided facts carry replayable certificates or witnesses, while unresolved entries remain typed outcomes rather than a global verdict.

2.2 Structural Comparison and Certifying Verification

Structural model comparison. ORGEval compares optimization models by graph-theoretic canonicalization (Wang et al. 2025). Structural matching is efficient but can reject pairs whose feasible sets coincide after a verified change of representation; our $E2$ relation certifies exactly this same-space, mapped feasible-set equality without claiming full-profile equivalence.

Proof certificates and exact verification. Certifying algorithms attach independently checkable evidence to their answers, from proof-carrying code (Necula 1997) to certified integer-programming reasoning (Cheung, Gleixner, and Steffy 2017; Hoen et al. 2024; van Doornmalen et al. 2023; Bogaerts et al. 2022). Feasible-set containment rests on Farkas’ lemma and polyhedral theory (Schrijver 1986); projected polyhedra and extended formulations are studied by Yannakakis (1991), Conforti, Cornuéjols, and Zambelli (2013), Kellner (2015), and Liberti (2009). Exact rational LP/MIP methods (Aplegate et al. 2007; Cook et al. 2013) make the resulting evidence checkable without additional floating-point error during verification. We adapt these tools into MODELEQUIVBENCH, in which every decided fact is independently re-verified using exact arithmetic.

3 The ModelEquivBench Framework

3.1 Models, Evidence Policy, and Outcome States

Models. Let M have n_M decision variables collected in $x = (x_1, \dots, x_{n_M})^\top$, with per-coordinate domains $D_{M,i}$ (continuous, integer, or binary) and $D_M = \prod_i D_{M,i} \subseteq \mathbb{R}^{n_M}$. Its feasible set is $F_M = \{x \in D_M : A_{\text{ub}}x \leq b_{\text{ub}}, A_{\text{eq}}x = b_{\text{eq}}\}$; it has linear objective $f_M(x) = \text{obj}_M^\top x$, sense in $\{\min, \max\}$, optimal value $\text{opt}(M)$ in the extended reals, and optimizer set $\text{argopt}(M)$. Empty-set conventions apply, and all coefficients are stored as exact rationals. We write R, C for reference and candidate, with ingested feasible sets F_R, F_C (bounds folded into inequalities). MODEL EQUIV-BENCH derives all claims from the models, never from declared labels.

Global disciplines. Four rules govern every dimension: (i) positive facts carry replayable traces, verified maps, or exact-rational certificates as appropriate; (ii) supported negatives carry failing traces or re-checkable witnesses; (iii) every fact from $E2$ onward is tied to a named, verified map φ ; and (iv) the evaluator is fail-closed—missing prerequisites, unsupported classes, extraction failures, and resource limits produce typed abstractions, never guessed FALSE outcomes. Unmet prerequisites are recorded separately as ABSENT.

Outcome states and definedness. A profile entry is not merely TRUE or FALSE. A dimension may also be UNKNOWN because the supported procedure established neither a positive conclusion nor a supported negative conclusion, UNKNOWN_RESOURCE because its resource budget was exhausted, UNKNOWN_UNSUPPORTED because the model lies outside the implemented envelope, or N/A because the relation is mathematically inapplicable. We reserve ABSENT for a different situation: an upstream prerequisite was not established, so the later dimension was never evaluated. For example, if no candidate model ingests, $E4$ is ABSENT, not N/A; if a same-space candidate has no auxiliary variables, $E3$ is N/A. This distinction matters both logically and statistically. A rate such as “ $E2$ decided given $E1$ ” conditions on the dimensions that were actually entered, whereas a cohort-wide count retains ABSENT cells in the accounting without treating them as failures of the relation. The resulting profile is therefore a partial, typed set of claims rather than a seven-bit vector.

3.2 Construction and Representation Alignment

$E0$: construction and exact ingestion. $E0$ asks whether the candidate code executes and yields a structurally valid exact-rational model that ingests. It is TRUE with a re-runnable build trace, FALSE with the failing parse/build/execution trace, UNKNOWN_RESOURCE when the execution cap is exceeded, and UNKNOWN_UNSUPPORTED for a valid but out-of-envelope construct (e.g. a quadratic objective). $E0$ does

not require a solver result: infeasible and unbounded models may still be $E0$ TRUE. $E0$ is established from ingestion and *checkpointed before* any later work, so a downstream timeout cannot corrupt it.

$E1$: verified representation alignment. Because a candidate may name and order variables differently, $E1$ searches for an admissible map from candidate to reference variables. The admissible family is $\Phi_{\text{adm}} = \Phi_{\text{same}} \cup \Phi_{\text{lift}}$, where Φ_{same} contains type-compatible coordinate permutations composed with per-coordinate transforms—identity, binary complement $x \mapsto 1 - x$ (0–1 only), and sign negation $x \mapsto -x$ (free continuous only)—and Φ_{lift} contains affine projection/section pairs with injective sections, relating a lower-dimensional reference to a higher-dimensional candidate. Candidate maps are proposed by a deterministic grammar (exact and original name matching, type-compatible permutation, signature matching, sign, complement, and affine lift) and each is checked for admissibility (dimension match, permutation bijectivity, per-transform legality by variable type, section injectivity, and the exact right-inverse identity $\pi \circ \sigma = \text{id}$ for affine lifts). $E1$ is TRUE if at least one admissible φ is verified and UNKNOWN if none is found in the searched family. Critically, $E1$ is *never* FALSE: absence of a found map is not proof that no map exists, so the outcome is coverage-limited, not a refutation.

3.3 Feasible-Set Relations

$E2$: same-space feasible-set relation. Fixing a verified same-space φ , $E2$ compares the mapped candidate feasible set $\varphi(F_C)$ with F_R as *subsets of the same semantic variable space* and classifies the pair as *equal* ($F_R = \varphi(F_C)$), *strict relaxation* ($F_R \subsetneq \varphi(F_C)$), *strict restriction* ($\varphi(F_C) \subsetneq F_R$), *incomparable*, or *unknown*. Each containment direction $P \subseteq Q$ is certified by exact nonnegative Farkas multipliers (μ, λ) with $a^\top = \mu^\top G + \lambda^\top H$ and $\beta \geq \mu^\top h + \lambda^\top h_{\text{eq}}$ for every row (a, β) of Q ; a failure to contain is witnessed by a feasible point of P violating a specific row of Q . This bidirectional, certificate-or-witness containment is the CMC engine; multipliers and witnesses are round-tripped through an independent verifier. Per-map relations are aggregated across the declared map universe: $E2$ is TRUE if *some* admissible φ is certified equal (and re-verified); it is FALSE_WITHIN_DECLARED_UNIVERSE only if that universe is *exhaustive* and *every* map in it separates with a re-verified witness (no equal, no unknown); otherwise it is UNKNOWN. The universe is declared exhaustive only when the type-compatible permutation family is fully enumerable ($n \leq 7$) and no binary coordinate is present, since binary complements are not exhaustively enumerated. Thus FALSE_WITHIN_DECLARED_UNIVERSE is a completeness-relative negative—“no admissible representation in the complete declared universe makes the feasible sets equal”—not a failed search.

Per-map and aggregated claims. The distinction between a relation under one named map and the aggregate

Dim.	Relation compared	Positive evidence / negative witness	Typed abstention
E0	Candidate code builds a structurally valid exact-rational model that ingests	Re-runnable parse/build trace; FALSE carries the failing trace	UNKNOWN_RESOURCE (exec. cap); UNKNOWN_UNSUPPORTED (e.g. quadratic)
E1	A verified admissible alignment $\varphi \in \Phi_{\text{adm}}$ exists (type/domain-compatible)	Explicit φ (permutation, complement, sign, or affine lift), re-checked exactly	UNKNOWN: no map found in the searched family (coverage-limited, never FALSE)
E2	Same-space feasible-set relation $\varphi(F_C)$ vs. F_R (equal / strict rel. / strict restr. / incomparable)	Two-way Farkas containment multipliers (equal); separating witness (FALSE_WITHIN)	UNKNOWN if no certified-equal and universe not exhaustive; UNKNOWN_RESOURCE
E3	Projected relation of a lifted candidate: $\pi(F_C)$ vs. F_R under section σ	Farkas obligations for $\pi(F_C) \subseteq F_R$, $\sigma(F_R) \subseteq F_C$, $\pi \circ \sigma = \text{id}$	UNKNOWN outside the affine-lift schema; N/A if no auxiliary vars
E4	Objective-order equivalence under φ	Identity/lift/complement objective identity; FALSE: two oppositely ranked points	UNKNOWN; N/A
E5	Optimal-value equality $\text{opt}(R) = \text{opt}(C)$	Two LP weak-duality sandwiches (LPs only; not used for integer models); implications from E2/E3	UNKNOWN; UNKNOWN_RESOURCE; N/A
E6	Optimizer-set bijection $\text{argopt}(R) \leftrightarrow \text{argopt}(C)$	Identity or 0–1 complement involution certificate (positive-only)	UNKNOWN (no certified-negative type); N/A

Table 1: The E0–E6 semantic profile. Each dimension is a distinct relation with its own certificate/witness type and typed abstention vocabulary. The dimensions are prerequisites for one another’s *definedness*, not rungs of a monotone equivalence ladder; a pair receives a profile, not a single label. N/A means a relation is mathematically inapplicable, whereas ABSENT means that an upstream prerequisite was not met. UR=UNKNOWN_RESOURCE; FALSE_WITHIN=FALSE_WITHIN_DECLARED_UNIVERSE.

E2 outcome is essential. A certified strict restriction under a particular φ does not rule out a second map that makes the sets equal. Hence a positive aggregate equality is existential, whereas a negative aggregate statement requires a complete finite universe and a re-verified separation for every member. When the universe is incomplete, the strongest sound conclusion after testing many non-equal maps is still UNKNOWN. This asymmetric policy deliberately sacrifices recall to prevent a failed representation search from being mislabeled as semantic inequality. Certificates and witnesses retain the map identifier, the exact transformed constraints, and the verified obligation, so every aggregate claim can be decomposed into independently checkable per-map facts.

E3: projected feasible-set relation. When the candidate carries auxiliary variables, *E3* compares F_R with the *projection* of the higher-dimensional F_C . With an affine projection $\pi : \mathbb{R}^{n_C} \rightarrow \mathbb{R}^{n_R}$ and an affine section $\sigma(x) = Cx + d$, the checker certifies three obligations— $\pi(F_C) \subseteq F_R$, $\sigma(F_R) \subseteq F_C$, and $\pi \circ \sigma = \text{id}$ —each via verifier-accepted Farkas multipliers or exact affine identities. Together these certify $\pi(F_C) = F_R$: the reference set is exactly the affine projection of the candidate’s feasible set. Projections outside the supported affine-lift schema (integer-auxiliary elimination, non-affine projection, reference equalities) yield UNKNOWN, and same-space cases with no auxiliary variables are N/A. If a prerequisite such as *E0* or *E1* is missing, *E3* is ABSENT rather than N/A. We stress that *E3* is *not* a level above *E2*: *E2* compares aligned sets in one space, while *E3*

bridges spaces of different dimension; they are different relations, reported independently.

3.4 Objective and Solution Relations

E4–E6: objective and optimizer relations. Under a verified candidate-to-reference map φ ($\varphi = \pi$ for lifts), *E4* checks whether f_C and $f_R \circ \varphi$ induce the same ordering on candidate feasible points with consistent optimization sense. Positives use an exact identity in the identity/lift/complement schema (additive constants cancel in comparisons); negatives use two oppositely ranked feasible points. *E5* certifies *optimal-value equality* $\text{opt}(R) = \text{opt}(C)$ by two LP weak-duality sandwiches of equal value or by implications from *E2/E3* equality with a preserved objective; duality certificates are *refused for integer variables*, since a relaxation bound does not certify the integer optimum. *E6* certifies an *optimizer-set bijection* $\text{argopt}(R) \leftrightarrow \text{argopt}(C)$ for a map with a certified inverse on optimizers (identity under *E2*-equality with identical objective, or a 0–1 complement involution). *E6* is *positive-only* in the current envelope: no certified-negative type exists yet, so absence of a correspondence is UNKNOWN, never FALSE; and *E3*-equality does not yield *E6* because the affine section is only a right inverse. These non-implications ($E2 \not\Rightarrow E4$, $E4 \not\Rightarrow E5$, $E5 \not\Rightarrow E6$, and their converses) are exactly why the framework reports a profile rather than a level.

Why the relations cannot be collapsed. The profile separates logically independent phenomena. Two for-

mulations can have the same feasible set but opposite objective senses, making $E2$ true and $E4$ false. Distinct feasible sets can share one optimum value, so $E5$ does not imply $E2$. Likewise, equal optimal values do not identify the same optimizers, and an optimizer correspondence says nothing about non-optimal feasible points. Even when one relation entails another under additional premises, the evaluator records the premises and the resulting certificate rather than silently propagating an unqualified global label. This is also why a baseline that checks only execution, value, or structure cannot be treated as an oracle for the whole profile.

4 Certifying Implementation

4.1 Exact Ingestion and Certificate Generation

MODELEQUIVBENCH orchestrates a verification core; synthesis is never trusted, and every artifact is re-verified before a decided fact is reported. Candidate code is extracted deterministically and executed in isolation under a 30 s cap; it must export an LP or MPS file, which is ingested into an exact-rational internal model. Every serialized decimal token is converted directly to a rational, so certification is exact with respect to the LP/MPS coefficients and introduces no additional floating-point error. $E1$ map proposals come from the deterministic grammar above. Feasible-set containment uses an exact-rational LP oracle to search for Farkas multipliers (positive direction) or a separating feasible point (negative direction); bounded 0–1 systems are handled by exact enumeration. Affine-lift and primal–dual optimality certificates are synthesized similarly. Every synthesized object is then re-checked by an independent verifier—two-way containment for $E2$ equality, witness verification for separations, the affine-lift obligations for $E3$, and the weak-duality re-check for $E5$ —and only re-verified facts are reported; all emitted certificate and witness artifacts were independently re-verified in all three runs (Table 3); the reported denominator counts artifacts rather than unique cells. Evaluation is *dimension-level checkpointed*, so a per-dimension timeout yields a typed UNKNOWN_RESOURCE for exactly that dimension while preserving earlier results. The protocol enforces *no repair and no resampling*: a malformed or code-less response is a genuine $E0$ outcome, not an occasion to re-prompt.

Gurobi plays a strictly bounded role: it builds and exports the generated models and reads LP/MPS artifacts, and it backs the value-matching and structural (ORGEval-style) baselines used only for comparison. It does *not* certify any $E1$ – $E6$ conclusion; all framework certificates use the exact-rational oracle and the independent verifier. Handling of transport timeouts and character-decoding failures in the harness provides engineering robustness and is not part of the evaluation semantics.

4.2 Independent Verification and Replay

Independent checking boundary. Certificate construction and certificate acceptance are separate code paths. The constructor may use an exact LP oracle to search for multipliers or witnesses, but the verifier only receives the serialized model, map, and proposed evidence and recomputes the required rational identities and inequalities. It does not trust solver status, LLM text, cached verdicts, or floating-point tolerances. A failed check prevents the proposed fact from being reported as decided rather than downgrading it to an uncertified Boolean. This separation is the operational basis for our use of “certified”: a third party can replay the compact evidence without replaying the generation process or accepting the constructor’s control flow.

System interface and replay artifacts. MODELEQUIVBENCH takes a reference model and one generated candidate as its unit of evaluation and emits three linked products: the typed $E0$ – $E6$ profile, the evidence objects supporting each decided entry, and a provenance manifest recording the map identifier, resource outcome, and source artifact hashes. This interface separates model generation from semantic assessment: a new generator can be evaluated without changing the certifier, while an updated certifier can replay persisted candidates without another paid model call. Dimension-level checkpoints also make partial profiles stable under interruption—a timeout in $E5$, for example, does not erase a previously verified $E2$ certificate. The resulting artifact is therefore more informative than a final boolean: it states what was proved, under which representation, what remained unresolved, and which compact objects a third party must check to reproduce the claim. Supplement Section F provides construction/replay pseudocode and the evidence schema.

5 Experiments

5.1 Experimental Setup

Cohort. Bench4Opt contains 394 source records (Wang et al. 2025); in our paired representation these form 197 base problems. A ground-truth-only support audit—depending solely on properties of the reference artifacts, never on any LLM output—partitions them into a development pilot set (20), a formal primary cohort (173), an unsupported audit set (4, e.g. nonlinear structure outside the envelope), and a malformed-or-missing set (0). We report results *only* on the frozen 173-problem formal cohort (SHA-256 prefix e9ad2018). This reference-only partition was frozen before formal generation, so no LLM output or evaluator result influenced inclusion. The 20-problem pilot was used only for development and is excluded from formal results. Each base problem is presented under two paired natural-language conditions—*Structured* (legacy internal identifier full) and *Unstructured* (concise)—giving $173 \times 2 = 346$ cells per model. The two conditions are paired by base problem. Besides condition-specific rates, we derive base-level discordant counts,

exact McNemar tests, and paired bootstrap intervals as descriptive analyses of this frozen cohort; they are not population-level claims.

Models and protocol. We evaluate three snapshots served through the AutoDL OpenAI-compatible endpoint: `gpt-5.4`, `claude-sonnet-4-6`, and `Qwen3.5-397B-A17B`. For each cell we take *one* generation at temperature 0.0 with *no repair* and *no resampling*; the candidate is executed under a 30 s cap and the *E1–E6* evaluator under a uniform 120 s cap. Prompts, cohort, mapping family, certificate rules, and baseline semantics are identical across all three models. Every run uses the same frozen prompt contract requesting one self-contained program that exports `candidate.lp`. Provider/API errors before candidate creation are kept separate from genuine E0 failure.

Baselines and reporting. Alongside the MODEL-QUIVBENCH profile, we run two comparison baselines: a solver-based *value-matching* implementation and the ORGEval-style structural implementation used in our harness. We do not claim that the latter is the unmodified official ORGEval implementation. Every rate names its denominator, and UNKNOWN, UNKNOWN_RESOURCE, UNKNOWN_UNSUPPORTED, N/A, ABSENT, provider/API failure, and certified negatives are kept distinct throughout; UNKNOWN never means “incorrect,” N/A means the relation is mathematically inapplicable, and ABSENT means a prerequisite was not met. Supplement Sections G–J provide the complete cohort/protocol record, full outcome tables, paired statistics, failure taxonomy, and replay instructions.

Outcome accounting. We report both cohort-wide counts and conditional coverage. Cohort-wide rows retain all 346 cells and expose where upstream generation or ingestion prevents later evaluation. Conditional rows ask, among cells for which a dimension is defined, how often the evaluator decides it: *E1* coverage is conditioned on *E0-TRUE*, and the main *E2* decided rate on *E1-TRUE*. The paired analysis instead uses all 173 base problems per model and defines binary end-to-end indicators for E0 success, E1 map-found, and E2/E5/E6 decided. Typed unknowns, N/A, and absence remain visible in the source matrix and are never relabeled as certified FALSE.

Table 3 reports the formal profiles, while Table 2 summarizes relation-specific baseline discrepancies.

Diagnostic	GPT-5.4	Sonnet 4.6	Qwen3.5
Execution-success overestimation	49	35	25
ORGEval rejection despite E2 equality	25	8	18
Strict value-match false acceptance	0	0	0

Table 2: Relation-specific baseline diagnostics. An execution-success overestimation is an E0-TRUE cell with a certified negative on at least one supported relation. The second row counts structural rejections on pairs for which E2 certifies mapped feasible-set equality; it is an E2-specific disagreement, not necessarily a full-profile false rejection. The ORGEval implementation is the one used in our harness.

5.2 Stage-Wise Model Profiles

Execution does not settle semantics. GPT, Sonnet, and Qwen produce ingestible candidates for 334/346, 156/346, and 196/346 cells. Yet their E2 decided counts are only 114/277, 45/130, and 87/164 among E1-TRUE cells. Across the three runs, 49, 35, and 25 E0-TRUE cells carry a certified negative on at least one supported relation. An execution-only evaluator would count every one of these as successful.

The evaluated snapshots fail at different stages. GPT has high E0 coverage (96.5%), so most losses occur downstream. Sonnet has the lowest E0 coverage (45.1%) under the frozen output contract. Qwen lies between them: 196 cells are E0-TRUE, while among 329 successful API responses 131 are genuine E0 failures and 2 are unsupported; another 17 cells end in a provider/API error before E0 is defined. Conditional E1 coverage is almost identical across the three models (82.9%, 83.3%, and 83.7%), showing why generation, ingestion, and semantic coverage must be reported separately.

Conditional coverage changes the comparison. Among cells with a verified map, Qwen reaches an E2 decision on 87/164 (53.0%), compared with 114/277 (41.2%) for GPT and 45/130 (34.6%) for Sonnet. This does not define a global winner: the denominators exclude different upstream losses. Rather, it shows how a model with lower end-to-end ingestion can have higher conditional certifier coverage, a distinction hidden by one aggregate score.

5.3 Baseline Discrepancies and Certified Case

Exact certification exposes baseline blind spots. ORGEval returns `not_equivalent` on 25, 8, and 18 cells for which E2 certifies mapped feasible-set equality for GPT, Sonnet, and Qwen, respectively. These are relation-specific discrepancies: the structural baseline rejects the pair globally, whereas E2 certifies equality of the mapped feasible sets. This comparison does not by itself assert objective-order, optimal-value, or optimizer-set equivalence. No batch exhibits a strict value-match false acceptance, but value matching addresses only one scalar relation and supplies no evidence for feasible-set, objective-order, or optimizer-set claims.

Dimension / metric (denominator)	GPT-5.4	Sonnet 4.6	Qwen3.5
<i>173 base problems $\times$ {Structured, Unstructured} = 346 cells per model; identical SHA-256 cohort-hash prefix e9ad2018</i>			
E0 TRUE (/346)	334 (96.5%)	156 (45.1%)	196 (56.6%)
genuine E0 FALSE / unsupported / provider error	12 / 0 / 0	190 / 0 / 0	131 / 2 / 17
E1 map found (/E0-TRUE)	277/334 (82.9%)	130/156 (83.3%)	164/196 (83.7%)
E1 UNKNOWN	57	26	32
E2 decided (/E1-TRUE)	114/277 (41.2%)	45/130 (34.6%)	87/164 (53.0%)
E2 TRUE / FALSE_WITHIN	108 / 6	41 / 4	84 / 3
E2 UNKNOWN / UNKNOWN_RESOURCE	118 / 45	61 / 24	50 / 27
E3 decided	0	0	0
E4 decided	139	40	91
E5 TRUE / FALSE	107 / 42	36 / 31	79 / 22
E6 TRUE (positive-only)	70	16	58
Certificate/witness artifacts re-verified	358/358	123/123	250/250
120 s cap-hit cells (union over dimensions)	70	35	39

Table 3: Formal MODELEQUIVBENCH profiles under the same single-generation, temperature-0.0, no-repair protocol. Qwen3.5-397B-A17B’s 17 provider/API errors occurred before candidate creation and are reported separately from genuine E0 failure. FALSE_WITHIN abbreviates FALSE_WITHIN_DECLARED_UNIVERSE. Re-verification denominators count emitted dimension-verdict artifacts, not unique cells.

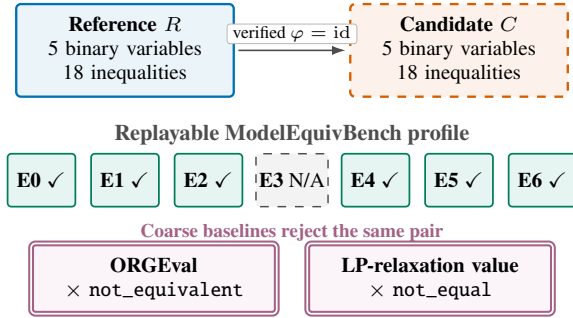


Figure 2: A replayable MODELEQUIVBENCH all-applicable-positive case (B4O_BASE_0031, Unstructured, Claude Sonnet 4.6). A verified identity map and exact enumeration certify E2; E4–E6 independently re-verify as true and E3 is N/A. Both coarse baselines nevertheless reject the pair.

A replayable all-applicable-positive case. Figure 2 shows a Sonnet case, B4O_BASE_0031 (Unstructured). An identity map aligns two models, each with five binary variables and 18 inequalities; exact enumeration certifies both E2 directions, and E4–E6 independently re-verify TRUE (E3 is N/A). ORGEval returns not_equivalent, while the LP-relaxation value baseline returns not_equal; the latter does not contradict E5, which concerns the original binary optima. The paired Structured cell has the same applicable profile, but ORGEval returns equivalent, illustrating representation sensitivity. Persisted paths and SHA-256 hashes make the case replayable.

5.4 Coverage, Abstention, and Reporting

Coverage and abstention are results, not errors.

The three runs contain 57, 26, and 32 E1-UNKNOWN cells; their E2 resource-timeout counts are 45, 24, and 27. At least one evaluator dimension hits the 120 s cap in 70, 35, and 39 cells. No formal cell receives a decided E3 outcome, so the study empirically exercises E0–E2 and E4–E6 while E3 records a zero-coverage boundary of the current affine-lift schema. The complement of a decided rate therefore mixes search incompleteness, unsupported structure, and resource abstention rather than forming a model error rate.

Auxiliary paired analysis. For GPT and Sonnet, base-level Structured–Unstructured comparisons show descriptive positive differences for downstream E1/E2 coverage, but no exact McNemar test survives Holm correction. We retain this as an auxiliary observation rather than a three-model ranking; all primary comparisons use the common aggregate denominators in Tables 2 and 3.

Soundness and coverage are separate axes. Every decided relation is backed by evidence that can be replayed independently, while coverage reports how often the current map family, supported schema, and resource budget reach a decision. High coverage without evidence risks confident but ungrounded labels; sound certification with limited coverage is transparent but incomplete. The profile reports both, preserving certified negatives, unresolved searches, resource limits, N/A, and upstream absence rather than forcing them into one score.

Implications for evaluator design. MODELEQUIVBENCH suggests that evaluation reports should separate at least three quantities that are often conflated: end-to-end generation success, conditional semantic coverage after ingestion, and the distribution of certified posi-

tive, certified negative, and abstaining outcomes within each relation. The same generator can look strong on the first quantity and weak on the second, or vice versa, as the three snapshots illustrate. A scalar “equivalence accuracy” cannot reveal whether errors arise from output contract violations, representation search, semantic disagreement, unsupported structure, or exhausted resources. Profile-level reporting also makes baseline comparisons relation-specific: value matching may be informative for $E5$ but silent about $E2$, while structural matching can disagree with a certified same-space feasible-set relation. In practical benchmark use, the profile can therefore serve both as a scorecard and as a debugging record, directing model or prompt improvements to the stage where evidence actually fails rather than to an undifferentiated final label.

A minimum auditable report should therefore pair the eligible cohort with end-to-end counts, per-relation conditional coverage, outcome distributions, re-verification rates, and the resource and map-completeness assumptions that govern UNKNOWN and FALSE_WITHIN_DECLARED_UNIVERSE.

Scope of claims. Results are limited to gpt-5.4, claude-sonnet-4-6, and Qwen3.5-397B-A17B on the frozen 173-base cohort, one generation per condition at temperature 0.0, no repair or resampling, and the 120 s evaluator policy. We claim neither universal superiority, state-of-the-art generation, nor completeness over all mathematical programs.

6 Limitations and Conclusion

Limitations. We evaluate three model snapshots with one generation per condition; broader claims require more models and repeated sampling. The supported envelope covers linear and bounded-discrete structure, while quadratic and general nonlinear models yield UNKNOWN_UNSUPPORTED. The finite $E1$ grammar is incomplete, $E3$ supports only the declared affine-lift schema, and $E6$ has no certified-negative type. Exact rational verification also incurs resource limits: under the 120 s policy, at least one dimension returns UNKNOWN_RESOURCE in 70, 35, and 39 GPT, Sonnet, and Qwen cells. Results may depend on provider-specific serving, and one sample per cell cannot quantify generation variance. The paired prompt analysis is descriptive—no McNemar comparison survives Holm correction—and the frozen cohort is not a random population sample. Finally, no formal cell has a decided $E3$ outcome, so empirical coverage of projected equivalence remains unestablished.

Conclusion. MODELEQUIVBENCH reports each candidate–reference pair as a certified $E0$ – $E6$ semantic profile rather than one opaque label. The frozen three-model study shows that execution is weak evidence of semantic correctness, different snapshots fail at different stages, and exact evidence with typed abstention makes both decisions and current coverage limits auditable.

References

- AhmadiTeshnizi, A.; Gao, W.; and Udell, M. 2024. OptiMUS: Scalable Optimization Modeling with (MI)LP Solvers and Large Language Models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, 577–596. PMLR.
- Applegate, D. L.; Cook, W.; Dash, S.; and Espinoza, D. G. 2007. Exact Solutions to Linear Programming Problems. *Operations Research Letters*, 35(6): 693–699.
- Bogaerts, B.; Gocht, S.; McCreesh, C.; and Nordström, J. 2022. Certified Symmetry and Dominance Breaking for Combinatorial Optimisation. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, 3698–3707.
- Cheung, K. K. H.; Gleixner, A.; and Steffy, D. E. 2017. Verifying Integer Programming Results. In *Integer Programming and Combinatorial Optimization: 19th International Conference (IPCO 2017)*, 148–160. Springer.
- Conforti, M.; Cornuéjols, G.; and Zambelli, G. 2013. Extended Formulations in Combinatorial Optimization. *Annals of Operations Research*, 204(1): 97–143.
- Cook, W.; Koch, T.; Steffy, D. E.; and Wolter, K. 2013. A Hybrid Branch-and-Bound Approach for Exact Rational Mixed-Integer Programming. *Mathematical Programming Computation*, 5(3): 305–344.
- Hoen, A.; Oertel, A.; Gleixner, A.; and Nordström, J. 2024. Certifying MIP-based Presolve Reductions for 0-1 Integer Linear Programs. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research (CPAIOR 2024)*, volume 14742 of *Lecture Notes in Computer Science*, 310–328. Springer.
- Huang, C.; Tang, Z.; Hu, S.; Jiang, R.; Zheng, X.; Ge, D.; Wang, B.; and Wang, Z. 2025. ORLM: A Customizable Framework in Training Large Models for Automated Optimization Modeling. *Operations Research*, 73(6): 2986–3009.
- Kellner, K. 2015. Containment Problems for Projections of Polyhedra and Spectrahedra. arXiv:1509.02735.
- Liberti, L. 2009. Reformulations in Mathematical Programming: Definitions and Systematics. *RAIRO – Operations Research*, 43(1): 55–85.
- Lu, H.; Xie, Z.; Wu, Y.; Ren, C.; Chen, Y.; and Wen, Z. 2025. OptMATH: A Scalable Bidirectional Data Synthesis Framework for Optimization Modeling. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, 40769–40802. PMLR.
- Necula, G. C. 1997. Proof-Carrying Code. In *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, 106–119.
- Ramamonjison, R.; Yu, T. T.; Li, R.; Li, H.; Carenini, G.; Ghaddar, B.; He, S.; Mostajabdaveh, M.; Banitalebi-Dehkordi, A.; Zhou, Z.; and Zhang, Y. 2022. NL4Opt Competition: Formulating Optimization Problems Based on Their Natural Language Descriptions. In *Proceedings of the NeurIPS 2022 Competitions Track*, PMLR, volume 220, 189–203.
- Schrijver, A. 1986. *Theory of Linear and Integer Programming*. John Wiley & Sons.
- van Doornmalen, J.; Eifler, L.; Gleixner, A.; and Hojny, C. 2023. A Proof System for Certifying Symmetry and Optimality Reasoning in Integer Programming. arXiv:2311.03877.
- Wang, Z.; Zhu, Z.; Li, Z.; Chen, C.; Han, Y.; Lin, Y.; Lin, Z.; Gu, A.; Hu, X.; Sun, R.; and Ding, T. 2025. ORGEval: Graph-Theoretic Evaluation of LLMs in Optimization Modeling. arXiv:2510.27610.
- Wei, A.; Cao, J.; Li, R.; Chen, H.; Zhang, Y.; Wang, Z.; Liu, Y.; Teixeira, T. S. F. X.; Yang, D.; Wang, K.; and Aiken, A. 2025. EquiBench: Benchmarking Large Language Models’ Reasoning about Program Semantics via Equivalence Checking. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 33868–33881. Association for Computational Linguistics.
- Xiao, Z.; Zhang, D.; Wu, Y.; Xu, L.; Wang, Y. J.; Han, X.; Fu, X.; Zhong, T.; Zeng, J.; Song, M.; and Chen, G. 2024. Chain-of-Experts: When LLMs Meet Complex Operations Research Problems. In *The Twelfth International Conference on Learning Representations (ICLR)*.
- Yannakakis, M. 1991. Expressing Combinatorial Optimization Problems by Linear Programs. *Journal of Computer and System Sciences*, 43(3): 441–466.
- Zhai, H.; Lawless, C.; Vitercik, E.; and Leqi, L. 2025. EquivaMap: Leveraging LLMs for Automatic Equivalence Checking of Optimization Formulations. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, 74288–74305. PMLR.

Supplementary Material

ModelEquivBench: Certifying Multi-Relational Evaluation of LLM-Generated Optimization Models

This technical appendix provides formal statements and proofs for the E0–E6 relations, certificate and witness schemas, algorithms for profile construction and independent verification, detailed cohort and protocol information, complete three-model outcome accounting, paired analyses, failure taxonomies, and reproduction instructions.

A Formal Setup, Scope, and Typed Outcomes

A.1 Optimization models and exact semantics

Definition A.1 (Optimization model). *A model M is a tuple*

$$M = (n, D, A_{\text{ub}}, b_{\text{ub}}, A_{\text{eq}}, b_{\text{eq}}, c, s),$$

where n is the number of variables; $D = D_1 \times \dots \times D_n$ with $D_i \in \{\mathbb{R}, \mathbb{Z}, \{0, 1\}\}$ (possibly intersected with exact rational bounds); $A_{\text{ub}}x \leq b_{\text{ub}}$ and $A_{\text{eq}}x = b_{\text{eq}}$ are exact-rational constraints; $c \in \mathbb{Q}^n$ is a linear objective; and $s \in \{\min, \max\}$ is the objective sense. Its feasible set and optimizer set are

$$F_M = \{x \in D : A_{\text{ub}}x \leq b_{\text{ub}}, A_{\text{eq}}x = b_{\text{eq}}\},$$

$$X_M^* = \underset{x \in F_M}{\text{argopt}} c^\top x.$$

The optimal value v_M^* is interpreted in the extended reals. For minimization, $v_M^* = +\infty$ when $F_M = \emptyset$ and $v_M^* = -\infty$ when the objective is unbounded below; the signs are reversed for maximization.

The reference model is denoted R and the generated candidate C . Every numeric token serialized in LP/MPS is converted directly to a rational. Thus, all certificate checks are exact with respect to the serialized model; the claim does not reconstruct real-valued quantities that may have been rounded before serialization.

A.2 Profiles are partial typed claim sets

The output for one candidate/reference pair is

$$\mathcal{P}(C, R) = (E0, E1, E2, E3, E4, E5, E6),$$

but this tuple is not a seven-bit vector and the indices are not a strength ranking. Each component records a relation, its applicability, its evidence, and a typed reason if no decision is returned.

A.3 Evidence discipline and prerequisite DAG

The framework obeys four global rules.

1. Every positive semantic fact is accepted only after exact independent re-verification of a certificate.

State	Meaning
TRUE	The stated positive relation is supported by evidence that passes the independent exact verifier.
FALSE / relation label	A supported negative relation has an explicit re-checkable witness. For E2 the aggregate negative is FALSE_WITHIN_DECLARED_UNIVERSE, not an unrestricted universal claim.
UNKNOWN	The relation is applicable but the supported search did not establish either side.
UNKNOWN_RESOURCE	The relation is applicable but a declared resource cap was reached.
UNKNOWN_UNSUPPORTED	A valid ingested object uses structure outside the implemented certification envelope.
N/A	The relation is mathematically inapplicable (e.g., E3 for a same-space candidate with no auxiliaries).
ABSENT	An upstream prerequisite was not established, so the relation was never entered.

Table 4: Typed outcome vocabulary. N/A and ABSENT are deliberately distinct.

2. Every supported negative fact carries a concrete witness.
3. E2–E6 are conditioned on a named verified representation map produced by E1.
4. Missing prerequisites, unsupported structure, failed evidence construction, and resource exhaustion never become guessed negatives.

E0 gates ingestion. E1 gates map-conditioned relations. E2 and E3 ask different feasible-set questions and do not form consecutive rungs. E4–E6 may be considered once an admissible map exists, but none is silently inferred from a lower index without recording the additional premises and evidence.

B E0 and E1: Ingestion and Representation Alignment

B.1 E0: candidate construction and exact ingestion

Definition B.1 (E0). *E0 asks whether a model response yields a deterministic candidate program that executes within the declared cap, exports an LP/MPS model, and is ingested into a structurally valid exact-rational model. E0 is TRUE with an independently replayable trace; it is a genuine FALSE when a successful model response contains no usable program, fails deterministically, or exports an invalid model. A provider/API failure before a candidate exists leaves E0 ABSENT. A valid but unsupported model is UNKNOWN_UNSUPPORTED, and an execution-cap event is UNKNOWN_RESOURCE.*

This definition separates three questions that are frequently conflated: provider delivery, candidate validity,

and semantic correctness. E0 TRUE asserts only the second. It makes no claim about E1–E6.

Proposition B.2 (E0 trace replay). *If the independent replay harness re-executes the recorded candidate program under the specified environment and obtains the same normalized LP/MPS-derived exact model, then the E0 positive claim is independently reproducible.*

Proof. The trace specifies the candidate program, execution command, resource cap, exported model, and deterministic exact-ingestion procedure. Replaying these objects reconstructs the same normalized tuple in Definition A.1; the replay decision does not rely on a solver’s semantic judgment. $\square$

B.2 E1: admissible mappings

A mapping aligns semantic decision variables before any relation is tested. The general admissible schema contains:

1. type-compatible coordinate permutations, including identity;
2. binary complement $x_i \mapsto 1 - x_i$ for 0–1 variables;
3. sign negation $x_i \mapsto -x_i$ for free continuous variables;
4. a declared affine lift/section pair for the supported E3 schema.

The general mapping schema admits sign negation for free continuous variables. In the reported formal run, however, the instantiated proposer manifest is the finite family listed in Supplement Section F.2: stable-name match, original-name match, identity, signature permutation, binary complement, and affine lift. No standalone sign-negation proposal is instantiated, so E1 and aggregate E2 coverage do not quantify over such maps.

Definition B.3 (Same-space admissible map). *A same-space map is an affine bijection*

$$\varphi(x) = Px + q,$$

where P is a signed permutation matrix and q is an exact offset. A negative entry denotes either sign negation on a free continuous coordinate with zero offset, or binary complement on a 0–1 coordinate with unit offset. It is admissible only if dimensions, variable types, bounds, and every coordinate transform are legal and the inverse is exact. In the reported finite proposal family, negative entries are instantiated only by binary complements.

Definition B.4 (Lift admissibility). *For $n_C > n_R$, a supported lift consists of a projection $\pi : \mathbb{R}^{n_C} \rightarrow \mathbb{R}^{n_R}$ and an affine section $\sigma(x) = Cx + d$ satisfying $\pi \circ \sigma = \text{id}$ exactly, together with compatible domains for all mapped variables.*

Proposition B.5 (Why E1 has no aggregate false). *Failure to find an admissible map in a finite searched family does not prove that no semantics-preserving map exists outside that family. Therefore the sound aggregate outcome is UNKNOWN, not FALSE.*

Proof. The deterministic mapping search evaluates a declared subset of all possible bijections and affine encodings. A negative conclusion over that subset cannot quantify over unsearched maps. Since E2–E6 may become positive under an unsearched map, the only sound result is a coverage-limited abstention. $\square$

Each accepted E1 artifact records the mapping type, exact parameters, domain compatibility checks, and a stable identifier used by every downstream certificate and witness.

C E2: Same-Space Mapped Feasible-Set Relations

C.1 Directional containment certificates

Let

$$P = \{x : Gx \leq h, Hx = h_{\text{eq}}\}, \\ Q = \{x : Ax \leq b\}.$$

with exact rational data after domain handling. The certificate form follows Farkas’ lemma and standard polyhedral containment arguments (Schrijver 1986). Equalities may be carried explicitly or converted to paired inequalities where supported.

Proposition C.1 (Farkas row certificate). *For a row $a_i^\top x \leq b_i$ of Q , suppose there exist $\mu_i \geq 0$ and unrestricted λ_i such that*

$$a_i = G^\top \mu_i + H^\top \lambda_i, \quad b_i \geq h^\top \mu_i + h_{\text{eq}}^\top \lambda_i.$$

Then every $x \in P$ satisfies $a_i^\top x \leq b_i$.

Proof. For $x \in P$,

$$a_i^\top x = \mu_i^\top Gx + \lambda_i^\top Hx \leq \mu_i^\top h + \lambda_i^\top h_{\text{eq}} \leq b_i,$$

where the first inequality uses $\mu_i \geq 0$ and $Gx \leq h$. $\square$

Corollary C.2 (Containment). *If Proposition C.1 has a verified multiplier pair for every row of Q , then $P \subseteq Q$.*

Proposition C.3 (Separating witness). *If a rational point z is independently verified to satisfy all constraints and domains of P and violates at least one row of Q , then $P \not\subseteq Q$.*

Proof. The point z belongs to P but not Q , which directly refutes containment. $\square$

For bounded 0–1 systems, exhaustive enumeration of all declared assignments is an exact finite certificate. The replay verifier re-checks every assignment or a compact deterministic enumeration trace.

C.2 Per-map relation classification

Fix one verified same-space map φ and write $P = \varphi(F_C)$ and $Q = F_R$. Two directional checks yield:

- *equal*: $P \subseteq Q$ and $Q \subseteq P$;
- *strict relaxation*: $Q \subseteq P$ and a witness proves $P \not\subseteq Q$;

- *strict restriction*: $P \subseteq Q$ and a witness proves $Q \not\subseteq P$;
- *incomparable*: witnesses refute both directions;
- *unknown*: at least one required direction is unresolved.

Proposition C.4 (Soundness of the five-way per-map diagnostic). *Whenever all evidence required by one of the first four labels above passes the independent verifier, the corresponding set relation holds under the recorded map φ .*

Proof. Equality follows from antisymmetry of set inclusion. Each strict label combines one certified inclusion with a verified point in the set difference. Two verified set-difference points establish incomparability. $\square$

C.3 Aggregation across maps

Let $\mathcal{M} \subseteq \Phi_{\text{adm}}$ be the declared verified map universe.

Definition C.5 (Aggregate E2). *The aggregate E2 result is:*

- *TRUE* if some $\varphi \in \mathcal{M}$ has certified equality;
- *FALSE_WITHIN_DECLARED_UNIVERSE* only when $\mathcal{M}$ is declared complete and every member has a verified non-equality witness;
- *UNKNOWN* otherwise.

Proposition C.6 (Soundness of aggregate E2). *A TRUE aggregate means that a named admissible representation makes the feasible sets equal. A FALSE_WITHIN_DECLARED_UNIVERSE aggregate means no map in the complete declared universe makes them equal. Neither statement quantifies over maps outside the declared grammar.*

Proof. The positive case is existential and inherits the per-map equality proof. In the negative case, completeness of the finite declared universe plus a verified non-equality witness for every member proves the universal claim restricted to that universe. Without completeness, such a universal conclusion is invalid, so the framework returns UNKNOWN. $\square$

The implementation declares the simple type-compatible permutation universe complete only when it is exhaustively enumerable (currently $n \leq 7$) and no binary coordinate is present. Binary-complement subsets are not exhaustively enumerated, so any binary coordinate makes the declared universe incomplete. This conservative boundary prevents failed map search from being mislabeled as semantic inequality.

C.4 Degenerate feasible sets

The empty set obeys ordinary containment conventions: $\emptyset \subseteq Q$ for every Q , and $P \subseteq \emptyset$ only when $P = \emptyset$. A positive empty-set equality therefore requires evidence that both sides are empty. A solver status alone is insufficient; the accepted artifact must be replayable under the exact supported procedure. Unboundedness is not an E2 special case because E2 concerns sets, not objective values.

D E3: Projected Feasible-Set Equality Under an Affine Lift

E3 applies when the candidate has auxiliary variables and a supported affine projection/section pair has passed E1.

Proposition D.1 (Affine-lift equality). *Let $\pi : \mathbb{R}^{n_C} \rightarrow \mathbb{R}^{n_R}$ and $\sigma : \mathbb{R}^{n_R} \rightarrow \mathbb{R}^{n_C}$ satisfy:*

$$\pi(F_C) \subseteq F_R, \quad (1)$$

$$\sigma(F_R) \subseteq F_C, \quad (2)$$

$$\pi \circ \sigma = \text{id}_{\mathbb{R}^{n_R}}. \quad (3)$$

Then $\pi(F_C) = F_R$.

Proof. Equation (1) gives one containment. For any $x \in F_R$, Equation (2) gives $\sigma(x) \in F_C$, and Equation (3) yields $x = \pi(\sigma(x)) \in \pi(F_C)$. Hence $F_R \subseteq \pi(F_C)$, proving equality. $\square$

The two set-containment obligations are discharged using exact Farkas evidence or bounded enumeration, while the right-inverse identity is checked by exact matrix arithmetic. This is a deliberately restricted, certifying instance of projected-polyhedron and extended-formulation reasoning (Yannakakis 1991; Conforti, Cornuéjols, and Zambelli 2013; Kellner 2015; Liberti 2009). Notice that π need not be injective; the section σ is injective as a consequence of $\pi \circ \sigma = \text{id}$.

E3 returns N/A for same-space candidates with no auxiliary variables. It returns UNKNOWN outside the declared affine-lift schema, including unsupported integer-auxiliary elimination, non-affine projection, and reference-equality cases outside the implemented schema. If E0 or E1 is missing, E3 is ABSENT, not N/A. No formal-cohort cell satisfies the implemented E3 certificate schema; this is a zero-coverage boundary, not evidence against Proposition D.1.

E E4–E6: Objective Order, Optimal Value, and Optimizers

E.1 E4: objective-order equivalence

For a verified map pairing candidate point y with reference point $x = \varphi(y)$, define the strict and weak preference relations induced by the recorded objective sense.

Definition E.1 (E4). *E4 is positive when, for every paired feasible y_1, y_2 ,*

$$f_R(\varphi(y_1)) \preceq_R f_R(\varphi(y_2)) \iff f_C(y_1) \preceq_C f_C(y_2),$$

where $\preceq_M$ uses the sense of model M .

Proposition E.2 (Positive-affine objective certificate). *If the senses agree and, on paired feasible points,*

$$f_C(y) = \alpha f_R(\varphi(y)) + \beta \quad \text{for exact } \alpha > 0, \beta \in \mathbb{Q},$$

then E4 holds.

Proof. A strictly positive affine transform preserves all weak, strict, and equality comparisons. The additive constant cancels in pairwise differences. $\square$

The implemented positive forms include identity, supported lifts, and binary complements that induce an exact positive-affine identity. A supported negative is a pair of feasible points whose rankings disagree, re-checked against both models and the named map.

E.2 E5: optimal-value equality

Definition E.3 (E5). *E5 asks whether $v_R^* = v_C^*$ as extended-real values under the original variable domains and recorded senses.*

Proposition E.4 (Primal–dual optimum certificate). *For a continuous rational LP, a primal-feasible point and a dual-feasible point with equal objective values certify the exact optimum by weak duality. If such certificates for R and C yield the same rational value, E5 is positive.*

Proof. Weak duality places every primal objective on one side of every dual objective. Equality of a feasible primal and dual pair closes the bound and proves optimality. Repeating this for both models and comparing the exact values proves E5. $\square$

For integer or binary variables, an LP-relaxation dual certificate does not certify the integer optimum and is refused. Bounded 0–1 cases may instead use exact enumeration. E5 can also be implied by a certified feasible-set equality and objective identity, but the report records those premises rather than silently transferring a global equivalence label.

E.3 E6: optimizer-set equivalence

Definition E.5 (E6). *E6 is positive when the recorded admissible map restricts to a bijection between X_C^* and X_R^* .*

Proposition E.6 (Optimizer transport). *Suppose φ is a bijection between F_C and F_R and E4 holds. Then $y \in X_C^*$ if and only if $\varphi(y) \in X_R^*$, so φ restricts to an optimizer-set bijection.*

Proof. If y is candidate-optimal, no feasible y' is strictly preferred. E4 transports this ordering to the reference set, and feasible-set bijectivity ensures every reference competitor has a candidate preimage. Therefore $\varphi(y)$ is reference-optimal. The reverse direction follows from φ^{-1} . $\square$

E6 accepts positive certificates only for forms with an exact certified inverse on optimizers (identity and the supported 0–1 complement involution). The implemented schema has no certified-negative E6 type. Consequently, unsupported or unproved cases are UNKNOWN, never FALSE. E3 equality alone does not imply E6 because its section is only a right inverse and need not represent every candidate optimizer uniquely.

Non-implication	Counterexample sketch
$E2 \not\Rightarrow E4$	Same feasible set $[0, 1]$; one model minimizes x , the other maximizes x .
$E4 \not\Rightarrow E5$	Same feasible set and $f_C = 2f_R + 1$: order is preserved but raw optimum values differ.
$E5 \not\Rightarrow E6$	On $\{0, 1\}$, minimize x versus minimize $1 - x$: both optimum values are 0, but the optimizers differ under identity.
$E6 \not\Rightarrow E2$	Reference feasible set $[0, 1]$ and candidate feasible set $\{0\}$, both minimizing x , share optimizer 0 but not the full feasible set.
E3 is not above E2	E2 is a same-space relation; E3 is a projection relation across dimensions. Either may be inapplicable while the other is meaningful.

Table 5: Why the profile cannot be collapsed into a deepest passing level.

E.4 Explicit non-implications

F Certificate Construction, Independent Replay, and Algorithms

F.1 End-to-end evaluator

Prerequisite normalization follows one rule for every profile: an unmet prerequisite is ABSENT, whereas N/A is reserved for a relation that is mathematically inapplicable after all prerequisites hold. This distinction is used in Tables 9 and 10.

F.2 Deterministic E1 search and applicability

The evaluation uses no LLM map proposer. The implementation exhaustively runs its deterministic proposer list, de-duplicates identical maps, and checks each proposal structurally before it can support any downstream claim. The declared family evaluated for every E1 profile is

```
stable-name-match;
original-name-match; identity;
signature-permutation;
binary-complement; affine-lift.
```

Name matching yields either a same-space coordinate map or, when the candidate contains additional variables, a projection map. Identity and signature permutation require equal dimension. Binary complement is proposed only for pure binary same-space pairs and is then rechecked exactly. The affine-lift proposal records projected reference coordinates; E3 separately verifies the projection/section obligations.

E1 is TRUE if at least one proposal is admissible and UNKNOWN if the complete declared family yields none. It is never a certified negative. Once E1 is TRUE, E2 is applicable only to admissible same-space maps. Seven formal cells (two GPT-5.4 and five Claude Sonnet 4.6) contain an admissible affine-lift map but no admissible

Algorithm 1 Prerequisite-aware construction of one E0–E6 profile

Require: cell z , fixed provider configuration θ , cohort hash h

- 1: Load the response identified by fingerprint (z, θ) , or record provider ABSENT.
- 2: **if** no successful response exists **then**
- 3: Set E0–E6 to ABSENT; record provider status;
- 4: **return** .
- 5: **end if**
- 6: Extract at most one program; execute under the 30-s cap; ingest the exported LP/MPS exactly.
- 7: Set E0 from construction and ingestion.
- 8: **if** E0 is not TRUE **then**
- 9: Set E1–E6 to ABSENT; record the profile;
- 10: **return** .
- 11: **end if**
- 12: Exhaustively propose and exactly check every map in the declared E1 family.
- 13: **if** no admissible map is found **then**
- 14: Set E1 to UNKNOWN and E2–E6 to ABSENT; record the profile;
- 15: **return** .
- 16: **end if**
- 17: Set E1 to TRUE and run E2–E6 in order under one 120-s pair-level wall cap.
- 18: Serialize each proposed certificate or witness and independently replay it.
- 19: Emit a decided fact only if replay succeeds; otherwise emit the appropriate typed abstention.
- 20: Store checkpoints, hashes, evidence, baseline records, and the profile.

same-space map; their E2 state is therefore N/A rather than UNKNOWN. E3 then records whether the supported affine-lift schema can be certified.

F.3 Independent replay and evidence schema

The replay verifier reads neither natural-language prompts nor solver status strings. Each evidence record contains the cell and cohort identifiers, exact model hashes, map identifier, relation and directional obligation, rational multipliers/points/matrices/vectors, constructor status, verifier status, checkpoint timing, and source hashes. Reported re-verification denominators count dimension-verdict artifacts rather than unique cells; one cell can contribute multiple E4–E6 artifacts.

F.4 Exactness, support envelope, and timeout semantics

Gurobi is used for candidate construction/export, LP/MPS parsing support, and the external value and structural baselines. It is not the E1–E6 certifier. Exact acceptance uses rational arithmetic in the certifying tradition of Applegate et al. (2007); Cook et al. (2013). Continuous linear containment uses Farkas certificates or exact separating witnesses; bounded pure-binary

Algorithm 2 E2 relation under the complete declared same-space map universe

Require: exact models (M_C, M_R) and admissible E1 maps Φ_{adm}

- 1: $\Phi_s \leftarrow \{\varphi \in \Phi_{\text{adm}} : \varphi \text{ is same-space}\}$
- 2: **if** $\Phi_s = \emptyset$ **then**
- 3:
- 4: **return** N/A.
- 5: **end if**
- 6: **for each** $\varphi \in \Phi_s$ **do**
- 7: Normalize $\varphi(F_C)$ and F_R in a common exact coordinate system.
- 8: Certify $\varphi(F_C) \subseteq F_R$, or verify a separating point.
- 9: Certify $F_R \subseteq \varphi(F_C)$, or verify a separating point.
- 10: Store the per-map relation implied by the two verified directions.
- 11: **end for**
- 12: **if** some map is certified equal **then**
- 13:
- 14: **return** TRUE with that map and both directional certificates.
- 15: **else if** every map has a verified non-equality witness **then**
- 16:
- 17: **return** FALSE_WITHIN_DECLARED_UNIVERSE with the complete map manifest and witnesses.
- 18: **else if** the pair-level cap was reached **then**
- 19:
- 20: **return** UNKNOWN_RESOURCE.
- 21: **else**
- 22:
- 23: **return** UNKNOWN.
- 24: **end if**

cases use complete enumeration where declared. Unsupported general-integer containment, unimplemented lift schemas, and unavailable optimizer-set negatives produce typed abstentions rather than guessed booleans.

The 120-s limit is one hard wall-clock cap for the complete E1–E6 pair evaluator, not a fresh 120 s for every dimension. The evaluator records a checkpoint before each dimension. If the cap is reached, E0 and all completed dimensions keep their recorded outcomes; the in-progress dimension and every unreached successor are set to UNKNOWN_RESOURCE. Consequently, the cap-hit statistic is a union over cells, whereas the per-dimension resource counts are cumulative. Provider responses are identified by request fingerprints.

Algorithm 3 Independent replay of a proposed relation fact

Require: normalized model hashes, named map, relation claim, serialized evidence

- 1: Re-load the exact models and verify all source hashes.
- 2: Re-check map admissibility without consulting constructor booleans.
- 3: Dispatch on evidence type: Farkas multipliers, separating point, affine projection/section, objective identity or ranking witness, primal/dual optimum certificate, or optimizer bijection.
- 4: Recompute rational identities, domain membership, feasibility, strict violations, and right-inverse identities as applicable.
- 5: **if** every obligation holds exactly **then**
- 6:
- 7: **return** verified.
- 8: **else**
- 9:
- 10: **return** rejected; the proposed decided fact is removed.
- 11: **end if**

G Cohort, Models, Prompt Contract, and Statistical Protocol

G.1 Cohort construction and eligibility

Bench4Opt contains 394 Structured/Unstructured records, grouped into 197 paired base problems (Wang et al. 2025). Eligibility is determined from reference artifacts only. The 20-base pilot subset is defined by sorting canonical base indices and taking the first 20 reference pairs inside the declared linear envelope, skipping unsupported base 0016; this gives bases 0000–0015 and 0017–0020. For the remaining 177 bases, inclusion requires (i) both paired records, (ii) exact E0-successful ingestion of the reference LP, and (iii) membership in {continuous LP, bounded pure binary, general-integer MILP}. Candidate outputs play no role in selection. The formal stratum counts are 112, 29, and 32, respectively.

The formal cohort hash is

```
e9ad20181d447143a3b53b377426af88d
395753cfd6f733eac0872373376b31c
```

The accompanying reproducibility archive includes the 197-base eligibility record, the pilot list, the 173-base manifest, and the four exclusion reasons.

G.2 Model aliases and generation protocol

The exact template, model configuration files, request fingerprints, and credential-free provider-status records are included in the reproducibility archive. Provider aliases are reported exactly as used by the evaluation client. The provider did not expose immutable served checkpoint hashes, so model aliases and request fingerprints are the strongest available identifiers.

Partition	Base problems	Cells/model
Pilot subset	20	40
Formal primary cohort	173	346
Unsupported reference set	4	8
Malformed/missing	0	0
Total	197	394

Table 6: Disjoint and exhaustive Bench4Opt paired-base accounting. Only the 173-base formal cohort contributes to primary results.

Excluded base ID	Ground-truth-only reason
B4O_BASE_0016	quadratic objective
B4O_BASE_0028	quadratic constraint
B4O_BASE_0182	quadratic objective
B4O_BASE_0187	quadratic constraint

Table 7: Unsupported reference instances. No paired record is malformed or missing.

G.3 Software environment

The local evaluator and the ORGEval-style baseline implementation used in our harness run under Python 3.9.7 on Windows 11, build 26100, with Gurobi 10.0.2, NetworkX 2.6.3, and NumPy 1.26.4. Exact verification additionally uses Python’s `fractions.Fraction`. Provider-side hardware is not observable. CPU and RAM identifiers were not recorded in the profile metadata; the archive provides a replay environment specification and a system-information collector.

G.4 Provider-aware and prerequisite-aware accounting

Provider delivery status is reported separately from model semantics. A terminal provider/API failure before a response yields E0–E6 ABSENT. All 346 GPT-5.4 cells have successful responses. Claude Sonnet 4.6 has 337 successful responses and nine terminal read-timeout errors; among successful responses, 156 are E0 TRUE and 181 are E0 FALSE. Qwen has 329 successful responses and 17 terminal provider errors; among successful responses, 196 are E0 TRUE, 131 are E0 FALSE, and two are UNKNOWN_UNSUPPORTED.

Among the 190 Claude Sonnet 4.6 cells without an E0-TRUE candidate, 181 are candidate-level E0 FALSE outcomes and nine are provider absences. All E1–E6 denominators condition on the 156 E0-TRUE cells.

G.5 Statistical unit

Profile counts use the cell as unit; Structured–Unstructured comparisons use the 173 paired base problems. The paired analysis uses five binary end-to-end indicators: E0 success, E1 map found, E2 decided, E5 decided, and E6 decided. Table 12 reports exact two-sided McNemar tests and 10,000 base-level paired bootstrap resamples with seed 13. Holm correction covers all

Field	Setting
Provider/API	AutoDL; OpenAI-compatible chat completions
Model aliases	gpt-5.4; claude-sonnet-4-6; Qwen3.5-397B-A17B
Conditions	Structured (artifact key <code>full</code>); Unstructured (<code>concise</code>)
Samples	one generation per model/condition cell
Temperature seed	/ 0.0 / 7
Maximum output	4096 tokens
Repair/resampling	disabled
Output contract	one self-contained program exporting <code>candidate.lp</code> or MPS
Request time-out/retries	120 s; at most two retries
Candidate execution	30-s hard cap
Pair evaluator	/ 120-s hard cap / 120-s hard cap
ORGEval	
Fallback / substitution	disabled

Table 8: Generation and evaluation protocol. No model-specific prompt adaptation is used.

H Complete Formal Results and Baseline Outputs

H.1 Stage-wise outcome accounting

Metric	GPT-5.4	Sonnet	Qwen
Cells	346	346	346
Successful API responses	346	337	329
Provider/API error; E0 ABSENT	0	9	17
E0 TRUE	334	156	196
Candidate-level E0 FALSE	12	181	131
E0 UNKNOWN_UNsupported	0	0	2
E1 map found / E0 TRUE	277/334	130/156	164/196
E1 UNKNOWN	57	26	32
E2 decided / E1 TRUE	114/277	45/130	87/164
E2 TRUE	/ 108/6	41/4	84/3
FALSE_WITHIN_DECLARED_UNIVERSE			
E2 UNKNOWN / N/A	116/2	56/5	50/0
E2 UNKNOWN_RESOURCE	45	24	27
E3 decided	0	0	0
E4 decided	139	40	91
E5 TRUE / FALSE	107/42	36/31	79/22
E6 TRUE	70	16	58
Evidence artifacts independently verified	358/358	123/123	250/250
Cells hitting at least one 120-s cap	70	35	39
Worker termination/corrupt profiles	0	0	0

Table 9: Stage-wise summary after provider and prerequisite normalization. The model columns are GPT-5.4, Claude Sonnet 4.6, and Qwen3.5-397B-A17B. Conditional denominators are explicit; complements are not automatically semantic errors.

H.2 Complete typed outcome matrix

Table 10 is exhaustive: every dimension row totals 346. For Qwen, E4–E6 are ABSENT in 182 cells because E0 or E1 is not established. The corresponding E4/E5/E6 UNKNOWN counts are 40, 25, and 67, and the UNKNOWN_RESOURCE counts are 33, 38, and 39.

H.3 External baseline outputs

Baseline outcome	GPT-5.4	Sonnet 4.6	Qwen3.5
Value match: equal	169	54	108
Value match: not equal	76	62	51
Value match: abstain	89	40	37
Value match: absent (no E0-true candidate)	12	190	150
ORGEval: equivalent	168	84	110
ORGEval: not equivalent	163	71	84
ORGEval: unknown	3	1	2
ORGEval: absent (no E0-true candidate)	12	190	150
Execution-success overestimation	49	35	25
ORGEval rejection despite E2 equality	25	8	18
Strict value-match false acceptance	0	0	0

Table 11: Baseline outputs and relation-specific certified discrepancies. Baseline “absent” combines every cell without an E0-true candidate, including provider absence, because no baseline pair exists.

15 model-by-indicator tests. These analyses describe the evaluated cohort and do not establish universal prompt effects.

Model	Dim.	T	F	$F_{\mathcal{U}}$	U	U_r	U_u	N/A	A
GPT-5.4	E0	334	12	0	0	0	0	0	0
	E1	277	0	0	57	0	0	0	12
	E2	108	0	6	116	45	0	2	69
	E3	0	0	0	2	45	0	230	69
	E4	138	1	0	74	64	0	0	69
	E5	107	42	0	59	69	0	0	69
	E6	70	0	0	137	70	0	0	69
Claude Sonnet 4.6	E0	156	181	0	0	0	0	0	9
	E1	130	0	0	26	0	0	0	190
	E2	41	0	4	56	24	0	5	216
	E3	0	0	0	5	24	0	101	216
	E4	40	0	0	57	33	0	0	216
	E5	36	31	0	28	35	0	0	216
	E6	16	0	0	79	35	0	0	216
Qwen3.5-397B-A17B	E0	196	131	0	0	0	2	0	17
	E1	164	0	0	32	0	0	0	150
	E2	84	0	3	50	27	0	0	182
	E3	0	0	0	0	27	0	137	182
	E4	91	0	0	40	33	0	0	182
	E5	79	22	0	25	38	0	0	182
	E6	58	0	0	67	39	0	0	182

Table 10: Complete prerequisite-normalized typed outcome matrix over all 346 cells per model. $F_{\mathcal{U}}$ denotes FALSE_WITHIN_DECLARED_UNIVERSE; U_r , U_u , and A denote UNKNOWN_RESOURCE, UNKNOWN_UNSUPPORTED, and ABSENT. Every row sums to 346.

An ORGEval rejection despite E2 equality means that the harnessed ORGEval implementation returns `not_equivalent` while E2 certifies mapped feasible-set equality under a verified map. This is an E2-specific discrepancy, not a claim of complete profile equality. A strict value-match false acceptance requires value equality together with an E2-certified strict/incomparable feasible-set relation; none is observed.

I Paired Conditions, Case Study, and Failure Taxonomy

I.1 Structured versus Unstructured paired statistics

No row remains significant after the 15-test Holm correction. The smallest raw values are Claude Sonnet 4.6 E2 ($p = 0.0241$), GPT-5.4 E1 ($p = 0.0294$), and Qwen E1 ($p = 0.0488$); their adjusted values are 0.362, 0.412, and 0.634. The table therefore supports only descriptive coverage patterns.

I.2 Replayable certified case

For B4O_BASE_0031 under the Unstructured condition, the Claude Sonnet 4.6 candidate and reference each contain five binary variables and 18 inequalities. E1 verifies the identity map. Complete enumeration over all $2^5 = 32$ assignments independently verifies both E2 containments. The profile stores E1/E2 replay status in the map-level containment evidence, where both

directional checks are independently verified. E4, E5, and E6 also verify positive; E3 is N/A because no auxiliary variables exist. ORGEval nevertheless returns `not_equal`, illustrating why baseline labels must be tied to their precise relation. This is the case visualized in Main Figure 2.

I.3 Failure taxonomy

Analysis keeps the following categories disjoint: terminal provider/API error; successful response without an extractable program; execution/export failure; unsupported serialized model; no admissible E1 map in the complete declared family; certified semantic negative; unresolved semantic search; resource cap; mathematically inapplicable relation; and downstream absence after a failed prerequisite. This is why E0 failure, E1 coverage, semantic disagreement, and resource abstention are not collapsed into one error rate.

Model	Indicator	S	U	S-only	U-only	Δ pp [95% CI]	p	Holm p
GPT-5.4	E0 success	165	169	3	7	-2.3 [-5.8, +1.2]	0.344	1.000
	E1 map found	145	132	22	9	+7.5 [+1.2, +13.9]	0.029	0.412
	E2 decided	62	52	17	7	+5.8 [+0.6, +11.6]	0.064	0.767
	E5 decided	77	72	12	7	+2.9 [-1.7, +8.1]	0.359	1.000
	E6 decided	38	32	17	11	+3.5 [-2.3, +9.2]	0.345	1.000
Claude Sonnet 4.6	E0 success	77	79	38	40	-1.2 [-11.0, +9.2]	0.910	1.000
	E1 map found	68	62	39	33	+3.5 [-6.4, +13.3]	0.556	1.000
	E2 decided	29	16	21	8	+7.5 [+1.7, +13.3]	0.024	0.362
	E5 decided	38	29	21	12	+5.2 [-1.2, +11.6]	0.163	1.000
	E6 decided	11	5	8	2	+3.5 [+0.0, +6.9]	0.109	1.000
Qwen3.5-397B-A17B	E0 success	103	93	34	24	+5.8 [-2.9, +14.5]	0.237	1.000
	E1 map found	89	75	29	15	+8.1 [+0.6, +15.6]	0.049	0.634
	E2 decided	47	40	17	10	+4.0 [-1.7, +9.8]	0.248	1.000
	E5 decided	54	47	16	9	+4.0 [-1.7, +9.8]	0.230	1.000
	E6 decided	34	24	17	7	+5.8 [+0.6, +11.6]	0.064	0.767

Table 12: Base-paired Structured (S) versus Unstructured (U) descriptive analysis for all three models. Exact two-sided McNemar tests use discordant pairs; confidence intervals use 10,000 paired base-level bootstrap resamples with seed 13; Holm correction covers the 15 model-by-indicator tests.

J Reproduction Procedure and Artifact Contract

J.1 Reproducibility archive contents

The reproducibility code/data archive is organized as follows:

- `code/certequivbench/`: ingestion, mapping, relation construction, independent verification, base-lines, cohort eligibility, and reporting code;
- `artifacts/cohort/`: the 197-base eligibility record, 20 pilot IDs, 173-base manifest, and four unsupported records;
- `artifacts/profiles/`: all 1,038 per-cell profiles;
- `artifacts/candidates/`: candidate programs where available; normalized candidate models and their hashes are retained in the per-cell profiles; exported LP/MPS files were not retained;
- `artifacts/baselines/`: per-cell value and ORGEval records;
- `artifacts/provider_status/`: credential-free call status, model alias, fingerprint, retry, error category, and usage metadata;
- `analysis/`: normalization, table-generation, paired-analysis, and selected-case scripts plus their deterministic outputs;
- `environment/`: replay requirements, the recorded provenance, and a system-information collector.

Raw API keys, author identities, absolute personal paths, and provider response text are excluded. The third-party Bench4Opt corpus is not duplicated; the manifest supplies stable IDs and hashes, and the README explains how to place an authorized local copy for full re-ingestion.

J.2 Replay sequence and invariants

A reproducer first verifies the archive manifest and cohort hash, installs the specified environment, and places Bench4Opt at the documented local path. The replay then (i) executes the reference-support eligibility check, (ii) ingests the recorded candidate models without provider access, (iii) independently verifies all certificates and witnesses, (iv) rebuilds normalized profile and baseline summaries, (v) recomputes the 10,000-resample paired analysis with seed 13, and (vi) compiles the appendix and checklist. The process fails closed on a cohort hash, cell identity, model hash, evidence verdict, conditional denominator, provider-category partition, or cap-accounting mismatch.

J.3 Claim-to-evidence index

- **Profile is not a hierarchy.** Main Secs. 1, 3.1, and 3.4; Supp. A and E. Replay the non-implication examples and prerequisite DAG.
- **E2 equality is sound.** Main Sec. 3.3; Supp. C. Check two directional Farkas certificates or complete finite enumeration.
- **E2 aggregate negatives are scoped.** Main Sec. 3.3; Supp. C.3. Check the complete declared map manifest and a verified witness for every map.
- **E3 affine-lift equality is sound.** Main Sec. 3.3; Supp. D. Check projection/section matrices, two containments, and the exact right-inverse identity.
- **E4–E6 are distinct.** Main Sec. 3.4; Supp. E. Check objective identities or ranking witnesses, optimum certificates, and optimizer bijections.
- **Decided facts are independently checked.** Main Secs. 4.1–4.2; Supp. F. Replay serialized evidence with the independent verifier.

- **Three-model counts share one cohort.** Main Secs. 5.1–5.2; Supp. G–H. Check the cohort manifest/hash and per-cell profile matrix.
- **Baseline discrepancies are relation-specific.** Main Sec. 5.3; Supp. H–I. Join each baseline record to its E2/E4–E6 profile by cell ID.
- **Provider failures are not E0 failures.** Main Secs. 5.1–5.2; Supp. G.4 and H. Check provider-status records and the normalized typed-outcome matrix.
- **Paired prompt results are descriptive.** Main Sec. 5.4; Supp. G.5 and I.1. Recompute the paired matrix, seed-13 bootstrap, McNemar tests, and Holm correction.

K Interpretive Clarifications

Is MODELEQUIVBENCH a new dataset? No. It is an evaluator and certifying artifact system; Bench4Opt supplies the experimental instances.

Do E0–E6 define an equivalence ladder? No. E0 and E1 are prerequisites/alignments; E2 and E3 compare different spaces; E4–E6 ask distinct objective and solution questions.

Does E1 UNKNOWN mean non-equivalence? No. It means the complete finite declared family yielded no admissible map.

When is a state N/A rather than ABSENT? N/A requires satisfied prerequisites and mathematical inapplicability. For example, E2 is N/A when E1 found only an affine-lift map, and E3 is N/A when a same-space map leaves no auxiliary variables. If E0 or E1 fails, downstream relations are ABSENT.

Why is E3 decided on zero formal cells? No formal cell both required and satisfied the one implemented affine-lift certificate schema. This is a coverage result, not a counterexample to the E3 theorem.

Why can E5 be positive while a value baseline says not equal? The baseline may compare a solver-derived or relaxed scalar, whereas E5 is accepted only from exact relation-specific evidence for the declared domains.

Why can ORGEval reject an E2-equal pair? A structural graph comparison can distinguish alternative encodings even when exact mapped feasible sets coincide. The discrepancy is relation-specific.

Why only one generation per cell? The protocol evaluates one output per cell and excludes repair or best-of- k selection. It therefore does not estimate within-model sampling variance.

L Limitations, Machine-Readable Contract, and Responsible Use

The guarantees are conditional on exact ingestion, the named admissible map, the supported certificate envelope, and the correctness of the independent verifier.

They do not prove that the reference model captures the source natural-language intent. The mapping family is finite, E3 implements one lift schema, general-integer containment remains incomplete, and E6 has no certified negative type. Exact procedures may exhaust the pair-level resource cap. The empirical study uses three provider aliases, one fixed prompt, and one sample per condition; it is diagnostic rather than a universal ranking.

L.1 Machine-readable profile contract

A portable profile record exposes at least

```
cell_id, base_id, model_alias, condition,
cohort_hash;
provider_status, request_fingerprint,
candidate_status, e0_status;
e1_status, declared_map_family,
searched_maps[], admissible_maps[];
e2..e6: status, relation, map_id, reason,
evidence_id, verifier_status;
elapsed_seconds, pair_cap,
checkpoint, baselines, source_hashes,
environment_fingerprint.
```

Provider status is separate from E0. Each downstream entry carries its map and prerequisite context. Replay status is recorded either at the relation level or in the referenced map-level evidence object; the verifier resolves the referenced evidence before assigning the relation outcome.

L.2 Threats to validity

Construct validity. The reference is treated as ground truth. MODELEQUIVBENCH measures candidate-to-reference semantics, not whether either model faithfully captures the source problem.

Internal validity. Separate construction and replay paths, exact arithmetic, hashes, and fail-closed routing reduce but cannot eliminate implementation defects. The provider-aware normalization is deterministic and preserves the source records.

External validity. Conclusions are limited to the supported Bench4Opt cohort, the three served aliases, the fixed prompt/output contract, and the recorded software envelope.

Statistical conclusion validity. Primary tables are exact finite-cohort counts. Paired tests characterize the 173 bases and are not estimates of an open-ended model population.

L.3 Responsible use

The evaluator may expose generated code and optimization artifacts. Any shared archive must remove credentials, identities, absolute personal paths, and non-redistributable provider content. Certification against a reference model must not be presented as validation of a real-world deployment without separate checks of reference correctness, data provenance, operational constraints, and domain consequences.