BEYOND FORMAL CORRECTNESS: STRUCTURE-AWARE EVALUATION OF INFORMAL–FORMAL PROOF CORRESPONDENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Proof assistants such as Lean certify that a formal proof establishes its target theorem, but do not determine whether it follows the reasoning expressed in a supplied natural-language proof. We introduce Goal–Context Delta Alignment (GCDA), a reference-conditioned framework for measuring this proof-process correspondence. GCDA recovers natural reasoning-state transitions and their derived dependencies, and traces the formal proof through Lean goal states connected by tactic-induced Goal–Context Deltas. It aligns each informal transition with a connected formal transition region, then checks required-region coverage, whole-proof coherence, and dependency direction. We evaluate GCDA on ProofRouteBench, a benchmark built from ReasBook and miniF2F with 4,888 proof variants across 670 theorem groups. GCDA achieves MacroAUCs of 0.9398 on ReasBook and 0.8788 on miniF2F, with strong separation on explicit dependency-certificate and cycle changes.

1 INTRODUCTION

Large language models are increasingly combined with proof assistants such as Lean and Coq for formal theorem proving and autoformalization. A proof assistant elaborates a tactic script into a proof term and uses a small trusted kernel to check that the term has the type corresponding to the target theorem Barras et al. (1997); de Moura et al. (2015). A Lean-accepted proof therefore provides a rigorous guarantee of formal correctness. This guarantee applies to the formal statement and proof term submitted to the assistant. It does not establish that the intermediate claims, implicit premises, and dependencies of a preceding natural-language argument were encoded faithfully. For the same theorem, one Lean proof may explicitly realize the central lemmas and dependency route of the natural proof, whereas another may invoke a stronger library result, rely on broad automation, or follow a different but still correct strategy. The kernel accepts all valid alternatives without deciding which one corresponds to the supplied natural proof.

This distinction matters when natural-language proofs are audited. Formalizing a model-generated solution certifies only what was encoded in Lean. It does not retrospectively show that the original derivation contained no missing bridge, misused premise, or strategy substitution. Comparing the two proof processes can instead identify which natural steps receive formal evidence, which necessary content remains unexpressed, and where dependencies change. The same capability is relevant to textbook formalization, mathematical-solution review, and proof maintenance.

Common evaluation signals address different aspects of this problem. Formal theorem-proving systems are commonly assessed by verified solve rate or pass@k, which measures whether a model can find at least one kernel-accepted proof Zheng et al. (2022); Xin et al. (2024); Wang et al. (2024b); Ren et al. (2025); Wang et al. (2025a). Autoformalization evaluation focuses primarily on whether an informal statement and its formal translation express the same mathematical proposition Wu et al. (2022); Azerbayev et al. (2023); Gulati et al. (2024). EntailmentBank and ProofWriter represent reasoning through intermediate propositions and their dependencies, while ROSCOE and stepwise verification assess reasoning chains and individual steps Dalvi et al. (2021); Tafjord et al. (2021); Golovneva et al. (2023); Lightman et al. (2024). ProofFlow further evaluates semantic and

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.

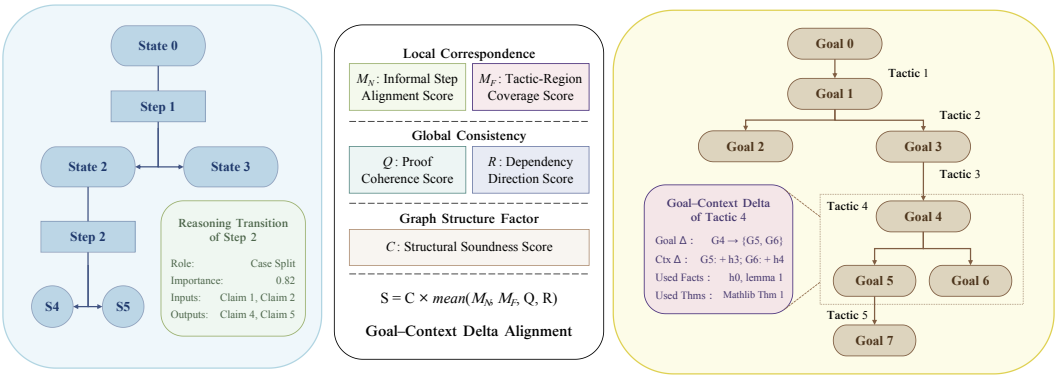

Figure 1: Overview of GCDA for proof-process correspondence.

dependency fidelity through ProofScore Cabral et al. (2026). Appendix A discusses this literature in greater detail.

We formulate the problem as *reference-conditioned proof-process evaluation*. Given a problem statement, a natural-language proof, a target theorem, and a verified Lean proof, the task is to measure how well the natural proof's central claims, obligations, dependencies, and route are supported by the specified formal process. A proof that follows another strategy may be mathematically correct yet have low correspondence to the chosen reference.

We propose *Goal–Context Delta Alignment* (GCDA). On the natural side, a language model recovers reasoning states, informal transitions, and derived dependencies between transitions. On the formal side, GCDA traces complete Lean goal states connected by tactic-induced transitions labeled with Goal–Context Deltas. It aligns each substantive informal transition with a connected tactic-transition region, allowing several tactics to realize one natural step. The score combines informal-step alignment M_N, tactic-region coverage M_F, proof coherence Q, dependency-direction consistency R, and structural soundness C.

We also introduce *ProofRouteBench* to evaluate proof-process metrics. The benchmark draws natural–formal proof pairs from ReasBook textbook and paper formalizations and miniF2F, retains three process-preserving controls, and constructs five process changes: required-bridge removal, invalid rewrite, wrong dependency, circular dependency, and alternative strategy. These families isolate missing necessary content, local mathematical errors, dependency corruption, circular support, and mathematically correct but route-different proofs. Our contributions are listed as follows.

- We introduce GCDA to measure agreement in proof logic and method between a natural-language proof and a specified verified Lean proof. It aligns natural reasoning steps with connected tactic regions using changes in goals and local facts, then checks support for the written steps, coverage of required formal steps, and consistency of their dependencies.

- We construct ProofRouteBench with 670 theorem groups and five controlled changes: required-bridge removal, invalid rewrite, wrong dependency, circular dependency, and alternative strategy. For each change type, we compute AUC by comparing process-preserving and changed proofs of the same theorem, then average these AUCs equally across eligible theorem groups.

- Under theorem-equal evaluation, GCDA outperforms a range of baselines on both corpora, with strong separation on explicit dependency-certificate and cycle changes, while retaining transition-, region-, and dependency-level evidence for inspection.

2 Task Definition

2.1 Input and Evaluation Object

Given a problem statement s, a natural-language proof p_N, a formal proof p_F, and a target theorem t, an evaluation instance x is defined as $x = (s, p_N, p_F, t)$. We restrict attention to instances $\mathcal{A}$ whose

problem goal is compatible with the formal theorem and whose formal proof has been accepted by the proof assistant:

$$\mathcal{A} = \{x \mid s \simeq t,\ p_F : t\}, \tag{1}$$

where $s \simeq t$ means that the informal goal and formal theorem express compatible target claims and assumptions, without asserting proof-process agreement; $p_F : t$ means that the proof assistant accepts p_F as a proof of t.

For $x \in \mathcal{A}$, GCDA measures how well the key claims, obligations, dependencies, and route of p_N are supported by p_F. The experimental score additionally conditions on the supplied dependency certificate D_N: $S_{\mathrm{GCDA}}(x; D_N) \in [0, 1]$ (Appendix G). This reference-conditioned score measures process correspondence, not absolute proof quality.

2.2 Desired Properties

Here a *route* is the choice of central lemmas, constructions, branches, and logical dependencies. We formulate three desiderata for a generic proof-process score $S : \mathcal{A} \to [0, 1]$.

Stability under process-preserving transformations. The same proof process may be expressed with different wording or notation, by locally merging or splitting steps, or by removing nonessential exposition. If a transformation τ preserves the key obligations, dependencies, and main route, the score should remain approximately stable:

$$|S(\tau(x)) - S(x)| \le \varepsilon_\tau. \tag{2}$$

We test this property with faithful paraphrases and nonrequired-step removal.

Bidirectional process evidence. High correspondence requires both formal support for the substantive steps stated in the natural proof and natural expression of the key obligations realized by the formal route. Consider two perturbations κ_{mis}, which replaces a natural step with a local derivation incompatible with the formal route, and κ_-, which removes a required bridge. An appropriate score should then satisfy

$$\begin{aligned} S(\kappa_{\mathrm{mis}}(x)) &< S(x), \\ S(\kappa_-(x)) &< S(x). \end{aligned} \tag{3}$$

Invalid rewrites and selected alternative strategies test the first direction. Required-bridge removal tests the second.

Sensitivity to route and structural changes. Replacing the principal strategy should reduce reference-conditioned correspondence even when theorem validity is preserved. Likewise, redirecting dependencies or introducing circular support should lower the score when most local content remains unchanged. For a route-changing transformation ρ and a dependency-corrupting transformation η, we expect

$$\begin{aligned} S(\rho(x)) &< S(x), \\ S(\eta(x)) &< S(x). \end{aligned} \tag{4}$$

Alternative strategy isolates mathematically correct but route-different proofs, while wrong dependency and circular dependency preserve most local transition content and alter its derived dependency structure. A route change must affect a central lemma, construction, induction structure, or governing proof principle, rather than merely wording or local granularity.

These properties define proof-process correspondence independently of GCDA. Section 4 tests them with process-preserving controls and five controlled process changes.

3 Goal–Context Delta Alignment

GCDA is transition-centric at the local alignment level and graph-aware at the route level. It represents both proofs as labeled state-transition structures, while the implementation serializes transitions as auditable records for matching and reachability. Appendix G gives model settings, prompts, and per-stage input/output formats. Final validation and obligation matching each receive one target natural-language proof at a time, together with its own intermediate records and the fixed formal evidence.

Algorithm 1: Natural-proof extraction and analysis

Require: Theorem statement and natural proof p_N.

1: **LLM extraction:** locate reasoning spans; recover available conditions, goals, and branch scopes without inserting unstated inferences.
2: Emit source-linked transitions T_N and endpoint states S_N; split or merge spans by their logical effects.
3: Recover dependencies D_N from claim use and branch scope, with supporting evidence and confidence.
4: **LLM analysis (Stage A):** annotate existing transitions with roles, expected effects, and importance α_i.
5: **Deterministic validation:** check fields, identifiers, source references, and numeric ranges; retain unresolved information.

Ensure: $G_N = (S_N, T_N, D_N)$ and annotated ProofUnit records.

3.1 Natural-Language Proof Graph

Given a natural-language proof p_N, GCDA recovers a state-transition structure

$$G_N = (S_N, T_N, D_N), \tag{5}$$

where S_N contains reasoning states, T_N contains informal reasoning transitions, and $D_N \subseteq T_N \times T_N$ contains derived dependencies between transitions. A reasoning state records the conditions and conclusions currently available, the branch or scope, a local objective when it can be recovered reliably, the associated source span, and extraction confidence. States are constructed conservatively: if the text jumps from A to C, GCDA does not invent an unexpressed intermediate state B.

Each transition $r_i \in T_N$ describes one local logical change within a branch. Its record stores the premises actually used, the produced conclusion, source and target reasoning states, scope, source span, confidence, and a proof role such as rewrite, reduction, witness construction, case split, bridge, induction step, or conclusion. It also receives a structural importance $\alpha_i \in [0, 1]$. Final conclusions, necessary bridges linking premises to later claims, and transitions opening or closing proof branches receive high importance; ordinary calculations and local rewrites receive intermediate values; purely expository material receives zero. Importance weights aggregation and prioritizes competing region assignments; it gives no direct local-match bonus.

A dependency $r_i \prec_N r_j$ is added to D_N when r_j uses the output of r_i, or when r_i opens a case or induction scope containing r_j. A transition may have several successors, and a later transition may require several predecessors. Weak textual order is retained only as auxiliary metadata unless explicit logical evidence supports a strong dependency. For computation, each r_i is materialized as an auditable ProofUnit record and D_N is stored as ordinary graph edges; this serialization does not change the state-transition semantics. Thus a ProofUnit represents an inference, not a state. The local matching term compares transitions and their endpoint states with formal regions, while D_N supplies the route-level evidence used by R, C, and the structural context of Q. Our natural-proof decomposition draws on the proof-state perspective of FaithSieve Wang et al. (2026c). Algorithm 1 summarizes extraction and subsequent analysis; Appendix B gives the details.

3.2 Lean State Deltas and Alignment

In formal reasoning, the same tactic can play different roles under different goals and local contexts, and a high-level tactic may perform several internal inferences. GCDA therefore uses the state-transition structure $G_F = (S_F, T_F, D_F)$. Each state in S_F is a complete Lean goal $g = (\Gamma \vdash \tau)$, where Γ is the local context and τ is the current target; goals with the same target but different contexts are distinct states. Each accepted tactic induces a transition $t_j \in T_F$ between its before and after goal states. We use Lean's Language Server Protocol (LSP) interface to extract source ranges, before/after goal sets $\mathcal{G}_j^-, \mathcal{G}_j^+$, observable fact uses, and subgoal structure for constructing tactic-transition records and their dependencies. Figure 2 illustrates these changes: T0 creates two goals, T1 introduces local variables, T4 creates induction branches, and the terminal markers indicate closed goals.

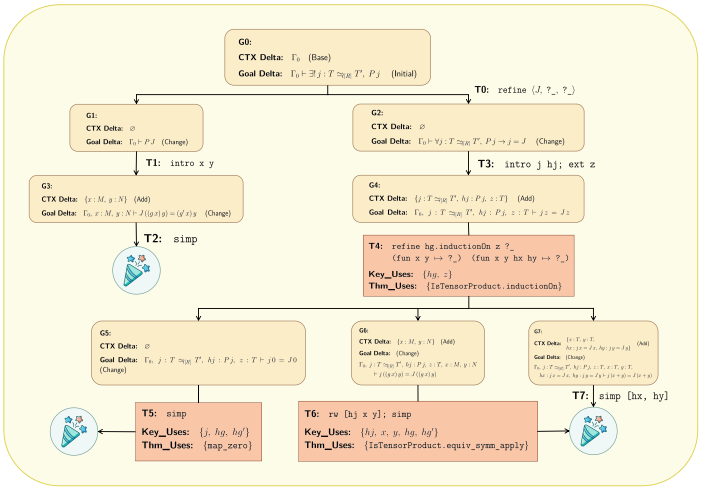

Figure 2: Lean goal-state transitions and connected tactic regions.

We summarize the transition evidence for tactic j as tactic-delta information $\mathcal{D}_j$, whose central component is the Goal–Context Delta $\Delta_j = (\Delta_j^{\text{goal}}, \Delta_j^{\text{ctx}})$, where

$$\begin{aligned} \Delta_j^{\text{goal}} &= (G_j^{\text{close}}, G_j^{\text{rewrite}}, G_j^{\text{new}}, G_j^{\text{keep}}, G_j^{\text{unknown}}), \\ \Delta_j^{\text{ctx}} &= (H_j^{+}, H_j^{-}, H_j^{\leftrightarrow}, H_j^{\text{keep}}, H_j^{\text{use}}). \end{aligned} \tag{6}$$

Each entry denotes transition evidence recovered by matching goals in $\mathcal{G}_j^-$ and $\mathcal{G}_j^+$. A before-goal with verified closure evidence is recorded in G_j^{close}, a matched goal with a changed target in G_j^{rewrite}, a newly generated successor in G_j^{new}, and a matched goal with an unchanged target in G_j^{keep}. Unreliable correspondences are placed in G_j^{unknown}. The context component records introduced, removed, changed, retained, and observably used entries. Goal–Context Delta is therefore an attribute of the tactic transition rather than a separate node. A `have` that preserves the target but introduces a reusable proposition remains substantive, whereas whether `simp` rewrites or closes a goal is determined by its transition evidence.

Tactic transitions are semantically directed hyperedges: a split may create several successor goals, while a close transition ends at a terminal marker. The derived relation $D_F \subseteq T_F \times T_F$ records reliable goal lineage, fact production and use, branch scope, and verified connector or containment relations; source order remains auxiliary. The implementation materializes each tactic transition as a Formal Delta Card and stores D_F between transition records. This transition-record serialization supports indexing, region selection, and reachability without introducing a second semantic graph.

Frozen formal requirements. A language model groups the theorem statement, Formal Delta Cards, and D_F into required regions. Each resulting *obligation* O_k records inputs, composed Delta, output, predecessor regions, and member IDs. After identifier and structural validation, the set $\mathcal{O}^+ = \{O_1, \ldots, O_K\}$ is frozen before reading natural variants and shared by all candidates for that theorem.

Region proposal and validation. For each informal transition r_i, the model proposes a connected region $\mathcal{R}_i = (\mathcal{C}_i, \mathcal{K}_i)$: core transitions $\mathcal{C}_i$ realize the effect; connectors $\mathcal{K}_i$ link its goal lineage or fact flow without earning independent semantic credit. The output cites formal IDs, composed Delta, inputs, and outputs. Shared facts may support several alignments, but do not themselves cover an omitted obligation. A deterministic validator checks IDs, connectivity, scope, and roles; final validation then returns the final partition and a judgment for the target proof, using only that proof's records and the fixed formal trace.

Compression is judged by agreement of observable inputs, state changes, outputs, and dependency boundaries; opaque tactics need not expose every internal inference. Coverage additionally requires

effect and output agreement, as specified in Section 3.3. Appendices C–D detail construction; Appendix I.8 gives a worked chain from natural records to final coverage.

Table 1: ProofRouteBench positive sources and ReasBook domain distribution.

Corpus or domain	Principal source families	Groups	ReasBook share
miniF2F	AMC, AIME, IMO, and MATH problems Zheng et al. (2022)	238	–
Real and functional analysis	Tao; Lebl Tao (2022); Lebl (2025)	67	15.5%
Convex analysis and optimization	Rockafellar; Nesterov Rockafellar (1970); Nesterov (2005)	62	14.4%
Algebra	Foundational Algebra; Magnus et al.; Serre Magnus et al. (2004); Serre (1977)	90	20.8%
Algebraic geometry	Stacks Project The Stacks Project Authors (2026)	47	10.9%
Probability	Klenke Klenke (2020)	39	9.0%
Topology and differential geometry	May; Lee May (1999); Lee (2013)	84	19.4%
Complex analysis	Cartan Cartan (1995)	43	10.0%
ReasBook-derived total	**Seven domains; twelve source families**	**432**	**100.0%**

3.3 Bidirectional Alignment and Scoring

Informal-step alignment. For an informal transition r_i and region $\mathcal{R}_i$, a language model judge compares the source-state conditions and formal inputs, the natural reasoning effect and composed Goal–Context Delta, and the produced conclusion and formal output. It also checks method, branch scope, and witnesses when applicable. The judge returns a candidate score in $[0, 1]$ together with exact Delta, output, and premise evidence. A deterministic calibration enforces the scoring rubric: an explicit contradiction forces zero; absence of principal Delta evidence caps the score at 0.45; matching the principal Delta without required-premise evidence caps the score at 0.70; and a score in $[0.90, 1]$ requires explicit output evidence. The resulting validated local score is m_i. Appendix E gives the complete anchors, output fields, and validation rules.

Recall that α_i denotes the structural importance of transition r_i. The informal-step alignment score is the importance-weighted mean

$$M_N = \frac{\sum_{i:\alpha_i>0} \alpha_i m_i}{\sum_{i:\alpha_i>0} \alpha_i}, \tag{7}$$

Necessary bridges and conclusion-producing transitions therefore contribute more than routine steps, while expository transitions do not enter the denominator.

Tactic-region coverage. For an informal transition r_i and frozen required region O_k, the judge assigns input, effect, output, and method scores in $[0, 1]$. Their fixed weighted combination defines the *region-match score*, which measures the local correspondence between one informal transition and one required tactic region:

$$s_{ik} = \lambda_{\text{in}} s_{ik}^{\text{in}} + \lambda_{\text{eff}} s_{ik}^{\text{eff}} + \lambda_{\text{out}} s_{ik}^{\text{out}} + \lambda_{\text{method}} s_{ik}^{\text{method}}, \qquad \sum_r \lambda_r = 1.$$

Effect and output agreement are required for genuine coverage. The *coverage credit* c_k is the best eligible informal-transition match for region O_k and measures the strongest valid coverage supplied by the entire natural proof:

$$c_k = \max_i s_{ik}\, \mathbf{1}[s_{ik}^{\text{eff}} \geq \tau_{\text{eff}},\ s_{ik}^{\text{out}} \geq \tau_{\text{out}}], \tag{8}$$

The tactic-region coverage score is defined as $M_F = \frac{1}{K}\sum_{k=1}^{K} c_k$. The fixed weights and thresholds are reported in Appendix E. For an incomplete natural proof with an omitted intermediate transition, the corresponding required region receives low or no coverage. This does not require the natural proof to restate bookkeeping tactics individually.

Whole-route semantics and structure. Locally valid pieces do not necessarily combine into a coherent proof. GCDA therefore validates each target proof as a whole against the fixed formal trace. This stage fixes the validated partition and returns a proof-specific coherence score Q based

on central claims, final support, missing bridges, case completeness, unsupported jumps, and contradictions. The validated transition–region scores define M_N, while Q records the whole-proof judgment. The dependency-direction score R then evaluates whether dependencies in D_N between aligned informal transitions are supported by directed reachability in D_F between their formal regions; it is not a general graph-similarity measure. A dependency with no observable directional evidence receives the neutral value $1/2$.

A valid proof process should not derive its support circularly. GCDA therefore introduces the structural-soundness factor C. For ordinary proofs with no strong dependency in a directed cycle, $C = 1$ and leaves the main score unchanged. If D_N^{cyc} contains the scored strong dependencies that participate in directed cycles, then

$$C = 1 - \frac{\sum_{d \in D_N^{\mathrm{cyc}}} q_d}{\sum_{d \in D_N} q_d}, \tag{9}$$

where q_d is dependency confidence. The final score is

$$\boxed{S_{\mathrm{GCDA}} = C \cdot \frac{M_N + M_F + Q + R}{4}}. \tag{10}$$

The four main components use a fixed equal-weight average, while C discounts explicit circular support. GCDA preserves transition-region alignments, required-region coverage, missing state changes, incorrect dependency directions, and cycle locations. As a result, a low score can be attributed to inconsistent local content, an incomplete process, an incorrect route relation, circular reasoning, or a valid but different strategy.

4 Numerical Evaluation

4.1 ProofRouteBench Construction

ProofRouteBench contains 670 theorem groups: 432 from ReasBook textbook and paper formalizations ReasBook Contributors (2026); Wang et al. (2026b) and 238 miniF2F problems Zheng et al. (2022); Table 1 gives the domain breakdown. Each group fixes an original natural–formal proof pair and adds faithful paraphrases and nonrequired-step removals as process-preserving controls. ReasBook/miniF2F contain $432/238$ originals, $432/238$ paraphrases, and $137/180$ removals.

Five controlled negatives test distinct process properties. *Required-bridge removal* deletes a necessary ancestor of the conclusion; *invalid rewrite* changes a local formula or conclusion; *alternative strategy* changes a principal lemma, construction, or proof principle while preserving correctness. *Wrong dependency* redirects a derived dependency, and *circular dependency* adds a descendant-to-predecessor edge. These two families edit the recovered graph certificate while preserving local content.

Edits apply only where the relevant transition or strong dependency exists and passes dependency and scope checks; ineligible groups omit the variant. All methods use the same frozen theorem-group population and eligible variants (Appendix F).

4.2 Evaluation Protocol and Baselines

For each controlled negative type, we compute the area under the receiver operating characteristic curve (AUC) within each theorem group using all available process-preserving positives. Ties receive half credit, and the resulting AUCs are averaged equally over valid theorem groups. We report the five type-specific theorem-equal AUCs, their arithmetic mean ($\mathrm{MacroAUC}$), and their minimum ($\mathrm{WorstAUC}$). Confidence intervals are obtained by cluster bootstrap over theorem groups. All methods are isolated from variant names, mutation targets, labels, expected rankings, and previous scores. GCDA evaluates each natural proof against the corresponding formal trace and returns a certificate and score for that proof. The external baselines score proof pairs independently under their declared input boundaries.

We compare four method families. *Textual and semantic metrics* include BLEU, chrF, ROUGE, BERTScore, BARTScore, MoverScore, E5 step F1, and ROSCOE-style scoring Papineni et al.

(2002); Popović (2015); Lin (2004); Zhang et al. (2020); Yuan et al. (2021); Zhao et al. (2019); Wang et al. (2022); Golovneva et al. (2023). *Statement-level adaptations* apply LeanScorer's semantic rubric Yu et al. (2026) and the E5-Mistral retrieval encoder used in LeanSearch Gao et al. (2024) to complete proof pairs; these are out-of-distribution adaptations. *Holistic judges* comprise Direct scalar, Direct rubric, and ProofScore Cabral et al. (2026). *Structural controls* comprise Holistic components without certificates and One-way structured alignment without frozen-region coverage. Appendix G gives prompts and implementation details. All 23 external score series use the same frozen miniF2F manifest of 1,783 variants, yielding 41,009 method–variant records.

4.3 Type-Specific Results

Table 2 reports scores and theorem-equal AUCs by process change. GCDA most reliably separates explicit cycle certificates, invalid rewrites, and wrong-dependency certificates. Bridge removal is harder because the remaining steps may still receive formal support. Alternative strategies are correct but route-different, so their scores measure route correspondence rather than error probability.

Table 2: GCDA results by process change.

	ReasBook		miniF2F	
Process change	Mean	AUC	Mean	AUC
Process-preserving (all)	**0.6507**	–	**0.4730**	–
Alternative strategy	0.3800	0.9618	0.3202	0.8431
Circular dependency	0.5058	0.9853	0.2496	0.9687
Invalid rewrite	0.4384	0.9691	0.2762	0.9237
Required bridge removed	0.5475	0.8541	0.3426	0.8197
Wrong dependency	0.5549	0.9288	0.3566	0.8390

Table 3: MacroAUC by corpus.

Method	ReasBook	miniF2F
BLEU	0.6157	0.6149
E5 step F1	0.5502	0.4102
LeanSearch-proof	0.5250	0.5334
LeanScorer-proof	0.6773	0.6199
Direct scalar	0.6848	0.6818
ProofScore	0.6719	0.6615
One-way structured	0.6462	0.6158
GCDA	**0.9398**	**0.8788**

The controls test stability rather than enforce it. On miniF2F, original, paraphrased, and nonrequired-step-removal proofs average 0.6080, 0.4081, and 0.3804, showing residual sensitivity to process-preserving changes. Theorem-equal AUC is 0.8840 (95% CI $[0.8681, 0.8998]$, 2,000 theorem-group bootstrap replicates). The paired ReasBook repeat analysis covers 165 groups and 1,214 matched rows in the current evaluation population. Within-theorem ordering agreement is 0.9177 and MacroAUC is $0.9404/0.9263$ across runs. These statistics characterize agreement between paired valid outputs; Appendix H gives the repeat and sensitivity analyses.

Table 3 compares corpus-level MacroAUC for the methods introduced above. Figure 3 aggregates type-specific results across corpora, weighting each by its eligible theorem groups; Appendix H reports the complete per-corpus matrices. Among the miniF2F baselines, Direct scalar is strongest with a MacroAUC of 0.6818, compared with 0.8788 for GCDA.

4.4 Ablation Studies

We further examine the contribution of each recorded component through leave-one-component-out sensitivity analyses. Figure 4 reports the available component-removal results and pools all eligible theorem groups from ReasBook and miniF2F. Each process change is therefore weighted by the number of theorems on which that variant is defined. The components capture different aspects of process correspondence: M_N aligns informal transitions with connected tactic regions. M_F tests whether frozen required tactic regions are covered by the natural proof. Q captures whole-proof coherence. R evaluates dependency direction among aligned regions. Finally, C discounts circular natural dependencies.

The ablations show category-specific tradeoffs. Removing M_N lowers miniF2F alternative-strategy AUC from 0.8431 to 0.7875, while improving bridge-removal AUC from 0.8197 to 0.8575. Removing M_F lowers ReasBook bridge-removal AUC from 0.8541 to 0.8254. These results are consistent with different roles for informal-transition support and required-region coverage, but do not establish that every component improves every category or the overall mean. In particular, removing

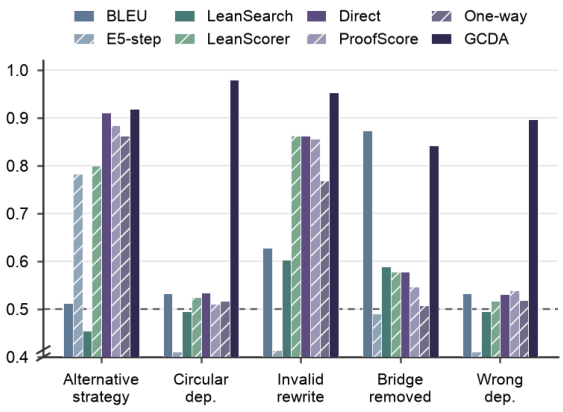

Figure 3: Theorem-equal AUROC by process change, weighted by eligible theorem counts.

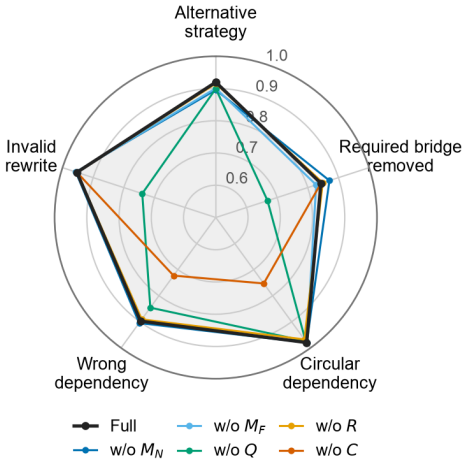

Figure 4: Component ablations: theorem-count-weighted AUROC over ReasBook and miniF2F.

M_N gives MacroAUC $0.9428/0.8825$ on ReasBook/miniF2F, compared with $0.9398/0.8788$ for full GCDA.

Removing Q substantially reduces invalid-rewrite and bridge-removal AUCs. Removing C primarily affects circular and wrong-dependency certificates; its apparent contribution on wrong dependency partly reflects edits that also introduce cycles. Removing R has smaller aggregate effects. These ablations deterministically recombine stored components; they do not rerun a pipeline with a stage removed.

4.5 Expert Assessment of Proof-Route Correspondence

Two experts assessed 50 anonymized natural-language–Lean proof pairs using a four-level rubric for agreement in core methods, key intermediate conclusions, and their dependencies. In the available numerical-review versions, their ratings agree on $43/50$ pairs (86.0%), with linear-weighted Cohen's $\kappa = 0.834$; all seven disagreements are one level apart. Grouping scores 2–3 as at least broadly corresponding and 0–1 as at most partially corresponding gives 92.0% agreement ($\kappa = 0.752$). Appendix H.9 details the rubric, rating distribution, and disagreement cases.

These statistics describe agreement after numerical review: one review was revised after access to the other reviewer's judgments, so they do not estimate independent blind inter-rater reliability. The results characterize agreement on proof-route judgments in this predominantly analysis-based sample. They do not establish GCDA–expert agreement, which requires a separate comparison of automatic scores with expert labels on matching inputs. Appendix H.9 also reports sensitivity to statement compatibility.

5 Conclusion

GCDA evaluates correspondence between natural-language and verified Lean proofs by aligning informal transitions with connected tactic regions and frozen formal requirements. On ProofRouteBench, it outperforms textual, statement-level, and holistic LLM baselines, especially on dependency and cycle perturbations. Its auditable evidence complements kernel verification by indicating whether the intended proof process is represented.

Reproducibility Statement

The paper specifies the metric, benchmark, and evaluation protocol. Appendices B–E detail graph extraction, Goal–Context Deltas, region construction, validation, and scoring; F gives corpus provenance; G documents the software environment, model settings, and prompts; H reports complete results and supplementary analyses. Code and data will accompany the submission.

AI Use Statement

In this work, we used generative AI tools for exploratory drafting, literature search and summarization, translation and language editing, conceptual and methodological discussion, assistance with implementation and code organization, organization and interpretation of experimental results, and the formatting of tables and figures. The authors made all final research and experimental decisions, reviewed all AI-assisted work, and independently verified the mathematical definitions, implementation, citations, and reported results. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

References

Zhangir Azerbayev, Bartosz Piotrowski, Hailey Schoelkopf, Edward W. Ayers, Dragomir Radev, and Jeremy Avigad. ProofNet: Autoformalizing and formally proving undergraduate-level mathematics. arXiv preprint arXiv:2302.12433, 2023. URL https://arxiv.org/abs/2302.12433.

Bruno Barras, Samuel Boutin, Cristina Cornes, Judicaël Courant, Jean-Christophe Filliâtre, Eduardo Giménez, Hugo Herbelin, Gérard Huet, César Muñoz, Chetan Murthy, Catherine Parent, Christine Paulin-Mohring, Amokrane Saïbi, and Benjamin Werner. The Coq proof assistant reference manual: Version 6.1. Technical Report RT-0203, INRIA, 1997. URL https://inria.hal.science/inria-00069968v1.

Alan Bundy. The use of explicit plans to guide inductive proofs. In *Automated Deduction—CADE-9*, volume 310 of *Lecture Notes in Computer Science*, pp. 111–120. Springer, 1988. doi: 10.1007/BFb0012826. URL https://doi.org/10.1007/BFb0012826.

Rafael Cabral, Tuan Manh Do, Xuejun Yu, Wai Ming Tai, Zijin Feng, and Xin Shen. ProofFlow: A dependency graph approach to faithful proof autoformalization. In *International Conference on Learning Representations*, pp. 58459–58483, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5fce5198dbf92d5ff45f74504431e2ff-Abstract-Conference.html.

Henri Cartan. *Elementary Theory of Analytic Functions of One or Several Complex Variables*. Dover Publications, 1995. URL https://store.doverpublications.com/products/9780486685434.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. Explaining answers with entailment trees. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7358–7370. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.585. URL https://aclanthology.org/2021.emnlp-main.585/.

Leonardo de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer. The Lean theorem prover (system description). In *Automated Deduction—CADE-25*, pp. 378–388. Springer, 2015. doi: 10.1007/978-3-319-21401-6_26. URL https://doi.org/10.1007/978-3-319-21401-6_26.

Guoxiong Gao, Haocheng Ju, Jiedong Jiang, Zihan Qin, and Bin Dong. A semantic search engine for Mathlib4. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 8001–8013. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.470. URL https://aclanthology.org/2024.findings-emnlp.470/.

Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=xYlJRpzZtsY.

Aryan Gulati, Devanshu Ladsaria, Shubhra Mishra, Jasdeep Sidhu, and Brando Miranda. An evaluation benchmark for autoformalization in Lean4. arXiv preprint arXiv:2406.06555, 2024. URL https://arxiv.org/abs/2406.06555. ICLR 2024 Tiny Papers track.

Jónathan Heras and Ekaterina Komendantskaya. Proof pattern search in Coq/SSReflect. arXiv preprint arXiv:1402.0081, 2014. URL https://arxiv.org/abs/1402.0081.

Xiaolin Hu, Qinghua Zhou, Bogdan Grechuk, and Ivan Y. Tyukin. StepProof: Step-by-step verification of natural language mathematical proofs. arXiv preprint arXiv:2506.10558, 2025. URL https://arxiv.org/abs/2506.10558.

Albert Qiaochu Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=SMa9EAovKMC.

Achim Klenke. *Probability Theory: A Comprehensive Course*. Springer, third edition, 2020. doi: 10.1007/978-3-030-56402-5. URL https://doi.org/10.1007/978-3-030-56402-5.

Ekaterina Komendantskaya, Jónathan Heras, and Gudmund Grov. Machine learning in proof general: Interfacing interfaces. In *Proceedings of the 10th International Workshop on User Interfaces for Theorem Provers*, volume 118 of *Electronic Proceedings in Theoretical Computer Science*, pp. 15–41. Open Publishing Association, 2013. doi: 10.4204/EPTCS.118.2. URL https://cgi.cse.unsw.edu.au/~eptcs/paper.cgi?UITP2012.2.

Jiří Lebl. Basic analysis I: Introduction to real analysis. https://www.jirka.org/ra/, 2025. URL https://github.com/jirilebl/ra/releases/tag/v6.2. Version 6.2.

John M. Lee. *Introduction to Smooth Manifolds*. Springer, second edition, 2013. doi: 10.1007/978-1-4419-9982-5. URL https://doi.org/10.1007/978-1-4419-9982-5.

Zenan Li, Yifan Wu, Zhaoyu Li, Xinming Wei, Fan Yang, Xian Zhang, and Xiaoxing Ma. Autoformalize mathematical statements by symbolic equivalence and semantic consistency. In *Advances in Neural Information Processing Systems*, volume 37, pp. 53598–53625. Curran Associates, Inc., 2024. doi: 10.52202/079017-1697. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/6034a661584af6c28fd97a6f23e56c0a-Abstract-Conference.html.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81. Association for Computational Linguistics, 2004. URL https://aclanthology.org/W04-1013/.

Wilhelm Magnus, Abraham Karrass, and Donald Solitar. *Combinatorial Group Theory: Presentations of Groups in Terms of Generators and Relations*. Dover Publications, 2004. ISBN 978-0-486-43830-6. URL https://openlibrary.org/isbn/0486438309.

J. Peter May. *A Concise Course in Algebraic Topology*. University of Chicago Press, 1999. URL https://press.uchicago.edu/ucp/books/book/chicago/C/bo3777031.html.

Erica Melis and Jörg Siekmann. Knowledge-based proof planning. *Artificial Intelligence*, 115(1):65–105, 1999. doi: 10.1016/S0004-3702(99)00076-4. URL https://doi.org/10.1016/S0004-3702(99)00076-4.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

Noor Islam S. Mohammad and Tamim Sheikh. The faithfulness gap: Certifying semantic equivalence between natural-language and formal mathematical statements. arXiv preprint arXiv:2606.16541, 2026. URL https://arxiv.org/abs/2606.16541.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005. doi: 10.1007/s10107-004-0552-5. URL https://doi.org/10.1007/s10107-004-0552-5.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318. Association for Computational Linguistics, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Maja Popović. chrF: Character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 392–395. Association for Computational Linguistics, 2015. doi: 10.18653/v1/W15-3049. URL https://aclanthology.org/W15-3049/.

ReasBook Contributors. ReasBook: Formalizations of mathematical textbooks and research papers in Lean 4. https://github.com/optpku/ReasBook, 2026. Accessed July 25, 2026.

Z. Z. Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, Z. F. Wu, Zhibin Gou, Shirong Ma, Hongxuan Tang, Yuxuan Liu, Wenjun Gao, Daya Guo, and Chong Ruan. DeepSeek-Prover-V2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. arXiv preprint arXiv:2504.21801, 2025. URL https://arxiv.org/abs/2504.21801.

R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970. URL https://press.princeton.edu/books/paperback/9780691015866/convex-analysis.

Jean-Pierre Serre. *Linear Representations of Finite Groups*. Springer, 1977. doi: 10.1007/978-1-4684-9458-7. URL https://doi.org/10.1007/978-1-4684-9458-7.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-acl.317. URL https://aclanthology.org/2021.findings-acl.317/.

Terence Tao. *Analysis II*. Springer, fourth edition, 2022. doi: 10.1007/978-981-19-7284-3. URL https://link.springer.com/book/10.1007/978-981-19-7284-3.

The Stacks Project Authors. The stacks project. https://stacks.math.columbia.edu, 2026.

George Tsoukalas, Jasper Lee, John Jennings, Jimmy Xin, Michelle Ding, Michael Jennings, Amitayush Thakur, and Swarat Chaudhuri. PutnamBench: Evaluating neural theorem-provers on the putnam mathematical competition. In *Advances in Neural Information Processing Systems*, volume 37, pp. 11545–11569. Curran Associates, Inc., 2024. doi: 10.52202/079017-0368. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/1582eaf9e0cf349e1e5a6ee453100aa1-Abstract-Datasets_and_Benchmarks_Track.html.

Haiming Wang et al. Kimina-Prover preview: Towards large formal reasoning models with reinforcement learning. arXiv preprint arXiv:2504.11354, 2025a. URL https://arxiv.org/abs/2504.11354.

Haocheng Wang, Baiyu Huang, Yingjia Wan, Xiao Zhu, Xiaoyang Liu, Yinya Huang, and Zhijiang Guo. FormalRx: Rectify and examine semantic failures in autoformalization. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026a. URL https://icml.cc/virtual/2026/poster/61085.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. arXiv preprint arXiv:2212.03533, 2022. URL https://arxiv.org/abs/2212.03533.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Ruida Wang, Jipeng Zhang, Yizhen Jia, Rui Pan, Shizhe Diao, Renjie Pi, and Tong Zhang. TheoremLlama: Transforming general-purpose LLMs into Lean4 experts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11953–11974. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.emnlp-main.667. URL https://aclanthology.org/2024.emnlp-main.667/.

Zichen Wang, Wanli Ma, Zhenyu Ming, Gong Zhang, Kun Yuan, and Zaiwen Wen. M2F: Automated formalization of mathematical literature at scale. arXiv preprint arXiv:2602.17016, 2026b. URL https://arxiv.org/abs/2602.17016.

Ziyu Wang, Bowen Yang, Chenyi Li, Yuan Zhang, Shihao Zhou, Bin Dong, and Zaiwen Wen. Translating informal proofs into formal proofs using a chain of states. arXiv preprint arXiv:2512.10317, 2025b. URL https://arxiv.org/abs/2512.10317.

Ziyu Wang, Qiming Dai, Yishan Wu, and Zaiwen Wen. Faithsieve: Fine-grained evaluation of math proofs with faithful formal evidence. arXiv preprint arXiv:2608.26310, 2026c. URL https://arxiv.org/abs/2608.26310.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. Autoformalization with large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 32353–32368. Curran Associates, Inc., 2022. doi: 10.52202/068431-2344. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/d0c6bc641a56bebee9d985b937307367-Abstract-Conference.html.

Huajian Xin, Daya Guo, Zhihong Shao, Zhizhou Ren, Qihao Zhu, Bo Liu, Chong Ruan, Wenda Li, and Xiaodan Liang. DeepSeek-Prover: Advancing theorem proving in LLMs through large-scale synthetic data. arXiv preprint arXiv:2405.14333, 2024. URL https://arxiv.org/abs/2405.14333.

Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. LeanDojo: Theorem proving with retrieval-augmented language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 21573–21612, 2023. doi: 10.52202/075280-0944. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/4441469427094f8873d0fecb0c4e1cee-Abstract-Datasets_and_Benchmarks.html.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822. Curran Associates, Inc., 2023. doi: 10.52202/075280-0517. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

Xuejun Yu, Jianyuan Zhong, Zijin Feng, Pengyi Zhai, Roozbeh Mohit, Wei Chong Ng, Haoxiong Liu, Ziyi Shou, Jing Xiong, Yudong Zhou, Claudia Beth Ong, Austen Jeremy Sugiarto, Yaoxi Zhang, Wai Ming Tai, Huan Cao, Dongcai Lu, Jiacheng Sun, Qiang Xu, Xin Shen, and Zhenguo Li. Mathesis: Towards formal theorem proving from natural languages. In *International Conference on Learning Representations*, pp. 91535–91559, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/93d5797656d3e70dca7535160a1458f3-Abstract-Conference.html.

Zhouliang Yu, Ruotian Peng, Keyi Ding, Yizhe Li, Zhongyuan Peng, Minghao Liu, Yifan Zhang, Zheng Yuan, Huajian Xin, Wenhao Huang, Yandong Wen, Ge Zhang, and Weiyang Liu. FormalMATH: Benchmarking formal mathematical reasoning of large language models. arXiv preprint arXiv:2505.02735, 2025. URL https://arxiv.org/abs/2505.02735.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. BARTScore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27263–27277. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/e4d2b6e6fdeca3e60e0f1a62fee3d9dd-Abstract.html.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 563–578. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1053. URL https://aclanthology.org/D19-1053/.

Xueliang Zhao, Wenda Li, and Lingpeng Kong. Subgoal-based demonstration learning for formal theorem proving. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 60832–60865. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhao24h.html.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. miniF2F: A cross-system benchmark for formal olympiad-level mathematics. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9ZPegFuFTFv.

A Related Work

A.1 Formal theorem proving and autoformalization

Interactive theorem provers such as Lean and Coq elaborate proof scripts into proof terms whose types are checked by a small trusted kernel de Moura et al. (2015); Barras et al. (1997). Recent language-model systems and benchmarks study whether a prover can construct a kernel-accepted proof for a given formal theorem, including miniF2F, LeanDojo, DeepSeek-Prover, TheoremLlama, DeepSeek-Prover-V2, Kimina-Prover, PutnamBench, and FormalMATH Zheng et al. (2022); Yang et al. (2023); Xin et al. (2024); Wang et al. (2024b); Ren et al. (2025); Wang et al. (2025a); Tsoukalas et al. (2024); Yu et al. (2025). Their principal endpoint is formal validity or theorem-solving success.

Autoformalization begins from informal mathematical content. Work on statement translation and its evaluation includes large-language-model autoformalization, ProofNet, the Lean4 autoformalization benchmark, and symbolic or semantic consistency checks Wu et al. (2022); Azerbayev et al. (2023); Gulati et al. (2024); Li et al. (2024). More recent diagnostic evaluations examine semantic drift and error localization at the theorem-statement level Mohammad & Sheikh (2026); Wang et al. (2026a); Yu et al. (2026). These tasks provide necessary validity and statement-correspondence signals, but they do not determine whether two complete proofs realize the same intermediate route.

A.2 Natural-language reasoning decomposition

A complementary literature represents reasoning as local steps, support relations, or proof plans. Classical proof planning separates high-level mathematical methods from low-level derivations Bundy (1988); Melis & Siekmann (1999), while EntailmentBank and ProofWriter make intermediate conclusions and entailment structure explicit in natural language Dalvi et al. (2021); Tafjord et al. (2021). Chain-of-thought, tree-structured reasoning, and mathematical step-verification work further motivate evaluating local claims rather than treating an explanation as one text block Wei et al. (2022); Yao et al. (2023); Lightman et al. (2024); Wang et al. (2024a); Hu et al. (2025).

Metrics such as ROSCOE evaluate multiple aspects of stepwise explanations, and factual-support metrics similarly decompose a response into locally checkable content Golovneva et al. (2023); Min et al. (2023). GCDA adopts the general lesson that intermediate reasoning should be represented explicitly. Its evaluation target differs, however, because support must be grounded in the state transitions and dependencies of a specified executable formal proof rather than in natural-language plausibility alone.

A.3 State- and graph-based proof representations

Proof representations have long used states, subgoals, plans, and graph structure. On the natural-language side, EntailmentBank and ProofWriter expose intermediate conclusions and entailment edges, while tree-structured reasoning and step verification organize claims into nonlinear or locally checkable structures Dalvi et al. (2021); Tafjord et al. (2021); Yao et al. (2023); Lightman et al. (2024). On the formal side, ML4PG extracts recurring patterns from proof traces; later systems use sketches or predicted subgoals to guide theorem proving, and Chain of States extracts intermediate formal states aligned with informal transitions before generating tactics between adjacent states Komendantskaya et al. (2013); Heras & Komendantskaya (2014); Jiang et al. (2023); Zhao et al. (2024); Wang et al. (2025b). Together, these works show that proof processes can be represented through intermediate states and their relations rather than only through surface text or tactic sequences.

GCDA adopts this representational view for evaluation rather than proof generation. Given a natural proof and a verified formal proof, it permits branching and many-to-many region alignment and derives dependencies from goal lineage, local-fact flow, and scope. This supports reference-conditioned comparison without imposing a linear chain.

A.4 Proof-level correspondence and evaluation

ProofFlow directly studies faithful autoformalization of complete proofs by converting an informal proof into a dependency graph and formalizing graph nodes as Lean lemmas; ProofScore combines node-level semantic agreement, compilation, and dependency agreement within that generated structure Cabral et al. (2026). This work demonstrates the importance of proof-level structure beyond theorem-statement equivalence. GCDA addresses a different evaluation setting in which the correspondence is not fixed by the generation pipeline: natural transitions may align with connected many-to-many tactic regions, while frozen formal obligations measure evidence in the reverse direction. Goal–Context Deltas and validated formal dependency paths supply the process evidence.

Textual, embedding, and holistic judging methods remain useful comparison families. BLEU, chrF, ROUGE, BERTScore, BARTScore, MoverScore, E5-based matching, and direct language-model judges capture complementary lexical or semantic relations Papineni et al. (2002); Popović (2015); Lin (2004); Zhang et al. (2020); Yuan et al. (2021); Zhao et al. (2019); Wang et al. (2022). The present benchmark asks whether such scores distinguish process-preserving variants from changes to bridges, dependencies, or strategy within the same theorem group.

B Natural State–Transition Construction

B.1 Reasoning states and transition records

The natural-language proof is represented semantically as $G_N = (S_N, T_N, D_N)$. A state $s_i \in S_N$ records the conditions and conclusions available at one local point of the argument, its branch or scope, a local objective when recoverable, source localization, and state-recovery confidence. A state is not a sentence or an inference action. State recovery is conservative: if the text moves directly from A to C, the extractor does not fabricate an unexpressed intermediate state B.

Each informal transition $r_i \in T_N$ connects source and target reasoning states and records the used conditions, produced conclusion, reasoning role, scope, importance, source span, and confidence. Principal roles include definition, assumption use, case split, intermediate bridge, theorem application, transformation, witness construction, and conclusion. Expository material may be retained for provenance but receives zero importance unless it expresses a mathematical transition.

The implementation serializes each r_i as a source-localized ProofUnit record. A sentence may produce several records when it contains distinct transitions, and adjacent sentences may be merged when they jointly express one inference. This representation makes transitions easy to index and validate; it does not change the semantic distinction between state nodes and transition edges.

Algorithm 2: Span-preserving natural-transition segmentation

1: Detect candidate mathematical clauses and scope markers.
2: Recover conservative reasoning states at observable transition boundaries.
3: Propose transition spans with used conditions, outputs, roles, and scopes.
4: Split a span only when each resulting transition has an independently interpretable effect.
5: Merge adjacent spans when their split would create no independent proof transition.
6: Attach endpoint-state IDs, source offsets, confidence, and unresolved fields.

Algorithm 3: Natural transition-dependency construction

1: **for** each ordered pair of transition records **do**
2: Add claim use when the later transition reuses the earlier output as an input.
3: Add scope dependence when a case or induction transition contains the later branch transition.
4: **end for**
5: Retain source order only as weak auxiliary metadata.
6: Validate direction, scope consistency, and unresolved-premise status.

B.2 Derived transition dependencies

The relation $D_N \subseteq T_N \times T_N$ does not introduce additional proof steps. It records dependencies among already extracted transitions. A claim-use dependency $r_i \prec_N r_j$ is added when r_j cites or reuses the output of r_i as an input. A scope dependency is added when a case split or induction transition opens the branch in which r_j occurs. A transition may have several dependent successors, and a later transition may require several predecessors.

For every candidate dependency, the extractor stores its type, direction, confidence, and supporting span or scope evidence. A premise that cannot be located reliably is marked unresolved rather than attached to the nearest sentence. Weak textual order may be retained for display and tie-breaking, but it is excluded from D_N unless explicit logical evidence supports the relation. The implementation stores D_N as ordinary edges between ProofUnit transition records.

B.3 Importance and validation

After the natural state–transition structure is fixed, all transitions in the same proof are analyzed jointly. For each r_i, the analysis records a structural importance $\alpha_i \in [0, 1]$, a within-proof rank, whether the transition has a logical function, a removal counterfactual, and its expected semantic state change. Importance is determined from the natural proof before formal alignment and without variant names or evaluation labels. Relevant evidence includes whether the transition produces a premise used later, opens or closes a branch, reduces the current objective, or is necessary for the stated route to reach its conclusion.

Importance measures structural contribution rather than sentence length, formula density, rhetorical emphasis, or mathematical correctness. Table 4 gives the fixed calibration. The scores are interpreted comparatively within a proof: transition role, position in D_N, and the removal counterfactual must support the resulting order.

Importance weights proof-level aggregation and prioritizes region allocation when natural transitions compete for the same nonshareable formal core. For a selected region, it gives no direct bonus to the evidence-based local match score m_i. The aggregation is

$$M_N = \frac{\sum_{i:\alpha_i>0} \alpha_i m_i}{\sum_{i:\alpha_i>0} \alpha_i}.$$

A high-importance transition therefore receives substantial credit only when supported by formal process evidence. Zero-importance records remain in the certificate for provenance but enter neither the numerator nor the denominator, preventing repetitive or expository additions from increasing the score. The frozen formal denominator used by M_F is independent of these natural-proof weights.

Validation covers both representation completeness and importance consistency. Every expected natural ID must occur exactly once, its logical type must belong to the declared inventory, and α_i

Table 4: Calibration of natural-transition importance.

Range	Structural interpretation
0	no logical function; explanatory, repetitive, or transitional content
$(0, 0.10]$	restatement or expansion with almost no independent contribution
$(0.10, 0.30]$	low-impact but genuine local inference
$(0.30, 0.60]$	substantive derivation, calculation, rewrite, or local goal reduction
$(0.60, 0.85]$	key intermediate claim, witness, major reduction, or controlling split
$(0.85, 1]$	final conclusion, indispensable bridge, or route-critical transition

Table 5: Natural transition records for the absolute-value example.

ID	Available source information	Logical type	State change	Information added to the target state
r_0	$x \in \mathbb{R}$; objective $\lvert x\rvert \geq x$	case split	split on $x \geq 0$ versus $x < 0$	positive- and negative-branch states
r_1	positive branch: $x \geq 0$	rewrite	unfold the definition of absolute value	$\lvert x\rvert = x$
r_2	$x \geq 0$ and $\lvert x\rvert = x$	derived fact	derive the branch inequality	$\lvert x\rvert \geq x$; positive branch closed
r_3	negative branch: $x < 0$	rewrite	unfold the definition of absolute value	$\lvert x\rvert = -x$
r_4	negative branch: $x < 0$	derived fact	derive the strict inequality	$-x > x$
r_5	$\lvert x\rvert = -x$ and $-x > x$	derived fact	combine the equality and inequality	$\lvert x\rvert \geq x$; negative branch closed
r_6	both branches establish $\lvert x\rvert \geq x$	conclusion	merge the two branch results	global conclusion $\lvert x\rvert \geq x$

must lie in $[0, 1]$; a zero-importance record cannot simultaneously be marked as having a logical function. Ranks are canonicalized deterministically from the continuous scores, with source order used only to break ties. The graph validator additionally checks source and target state IDs, source spans, used claims, produced conclusions, scope references, and the endpoints of every edge in D_N. Unsupported fields remain explicitly unresolved. Confidence and importance are distinct: an underspecified but route-critical bridge may have high importance and low recovery confidence, whereas a clearly written repetition may have high confidence and zero importance. If the minimum state–transition contract cannot be recovered, the evaluator abstains rather than interpreting structural uncertainty as dissimilarity.

B.4 Illustrative structures

Consider the statement $|x| \geq x$ for every real x, with the natural proof: "If $x \geq 0$, then $|x| = x$, so $|x| \geq x$. If $x < 0$, then $|x| = -x$ and $-x > x$, so again $|x| \geq x$. The two cases prove the claim." The initial state contains $x \in \mathbb{R}$ and the objective $|x| \geq x$. Table 5 makes the seven extracted transitions explicit.

A single sentence may therefore yield several records: the negative-case phrase $|x| = -x > x$ contains both a rewrite and a separate inequality derivation. Conversely, adjacent clauses are merged when neither expresses an independently meaningful state change.

After transition extraction, the relations in Table 6 are derived from claim use and branch scope. They are not inserted merely because the corresponding phrases occur in a particular textual order.

Although r_2 and r_3 are textually adjacent, they lie in different branches and r_3 does not consume the output of r_2, so D_N contains no edge $r_2 \prec_N r_3$. Similarly, r_3 and r_4 both use the negative-branch condition but can be performed independently. The example thus distinguishes textual order, branch scope, local state changes, and derived logical dependence.

Table 6: Derived dependencies in the absolute-value example.

Dependency	Type	Evidence
$r_0 \prec_N r_1, r_2$	scope	r_1 and r_2 occur in the $x \geq 0$ branch
$r_0 \prec_N r_3, r_4, r_5$	scope	r_3, r_4, and r_5 occur in the $x < 0$ branch
$r_1 \prec_N r_2$	claim use	r_2 uses the equality $\lvert x\rvert = x$ produced by r_1
$r_3 \prec_N r_5$	claim use	r_5 uses the equality $\lvert x\rvert = -x$ produced by r_3
$r_4 \prec_N r_5$	claim use	r_5 uses the inequality $-x > x$ produced by r_4
$r_2, r_5 \prec_N r_6$	branch merge	the global conclusion requires both branch conclusions

C Lean State–Transition and Region Construction

C.1 Goal-state identity and trace contract

The formal proof is represented semantically as $G_F = (S_F, T_F, D_F)$. A state is a complete Lean goal $g = (\Gamma \vdash \tau)$, where Γ is the local context and τ is the target. The target alone does not identify a state: equal targets under different local contexts remain distinct. Closed goals point to a terminal marker, and missing after-states remain explicitly unknown.

For each accepted tactic or action, the structured trace supplies a source range, before and after goal sets when available, local-context information, used constants and local facts, subgoal lineage, branch scope, and provenance. Tactic text alone is not formal process evidence. When an after-state is unavailable, the extractor does not infer closure from the tactic name.

C.2 Tactic transitions and Goal–Context Deltas

Each accepted tactic induces a transition $t_j \in T_F$. Its endpoints are the before and after goal states, and its payload includes the normalized effect type, Goal Delta, Context Delta, used facts and theorems, produced claims, subgoal lineage, scope, trace mode, and evidence quality. For transition t_j, let $\mathcal{G}_j^-$ and $\mathcal{G}_j^+$ be the before and after goal sets. Their partial correspondence is recovered from stable metavariable identities, target compatibility, and subgoal lineage.

The Goal Delta classifies preserved, rewritten, closed, introduced, and unknown goals. A rewrite preserves an obligation identity while changing its target; a new goal is linked to the transition that created it; closure is recorded only when verified. The Context Delta records introduced, removed, changed, retained, and observably used local facts. Goal–Context Delta belongs to the tactic transition because it describes the change between endpoint states.

Worked trace and the closure boundary. In the archived example of Appendix I.8, `intro x` exposes the pointwise periodicity goal and introduces `x` into the local context; the following `exact congrArg ... (hg x)` uses the periodicity hypothesis and closes that goal. The first Formal Delta Card therefore records the changed target and introduced variable, and the second records the used hypothesis and verified closure. The two cards retain their source actions and goal lineage. If an after-state were unavailable, absence of a successor in the packet would instead remain unknown: the word `exact` alone would not establish closure. The appendix presents the complete natural proof, Lean proof, and intermediate records for this same example.

C.3 Hyperedges and derived formal dependencies

Tactic transitions are directed hyperedges at the semantic level. A rewrite maps one goal state to one successor, a split maps one state to several successors, a close maps a state to the terminal marker, and a context-only transition may preserve the target while changing Γ. Multi-goal tactics may affect several obligations together. An incidence representation, goal state $\rightarrow$ transition record $\rightarrow$ successor goal state(s), provides an equivalent computational view.

The derived relation $D_F \subseteq T_F \times T_F$ records reliable causal relations between tactic transitions. Its evidence includes goal and subgoal lineage, production and later use of local facts, case or induction scope, and verified containment or connector relations. Source order is retained only as weak

Algorithm 4: Goal–Context Delta transition extraction

for each structured tactic transition t_j **do**
    Match before and after goals using metavariable identity, target compatibility, and lineage.
    Mark disappeared goals as closed only with verified closure evidence.
    Classify preserved, rewritten, introduced, and unknown goal changes.
    Compare local contexts and record fact introduction, change, retention, and use.
    Record one-to-many subgoal lineage, branch scope, and provenance.
end for
Export one Formal Delta Card per transition with evidence and confidence fields.

Algorithm 5: Frozen required-region construction

Build connected tactic regions from the reference transition structure.
Summarize each region by its inputs, composed Goal–Context Delta, output, and dependency boundary.
Merge only regions with compatible external boundaries and verified containment.
Validate a source transition and stable theorem-group identifier for every region.
Freeze the obligation set and its hash before reading any natural-proof variant.

metadata. A global theorem application is distinguished from a locally produced bridge so that route evidence is not inferred from names alone.

The implementation materializes each t_j as a Formal Delta Card and stores D_F as ordinary edges between cards. This transition-record serialization supports region selection, connectivity checks, and widest-path reachability without defining a second action-centered semantic graph.

D Connected Regions and Frozen Obligations

D.1 Connected tactic regions

For an informal transition r_i, a formal region is a connected set of tactic transitions that provides compatible input facts, carries out the principal state change, and produces the corresponding output. We write $\mathcal{R}_i = (\mathcal{C}_i, \mathcal{K}_i)$, where $\mathcal{C}_i \subseteq T_F$ contains core transitions and $\mathcal{K}_i \subseteq T_F$ contains connectors required by goal lineage or fact flow. Connectivity is checked in the transition incidence and dependency structure. Connector transitions may link a region's boundaries but receive no independent semantic credit; unrelated transitions are not absorbed merely because they occur earlier in the proof.

The region boundary is defined by observable inputs, composed Goal–Context Delta, outputs, and dependencies. This makes cross-granularity alignment explicit without treating every valid proof as the same route. A macro or opaque tactic can belong to a region when its external transition and dependency boundary are compatible with the informal transition. Opacity describes observability, not route identity.

D.2 Frozen required tactic regions

Before any natural-proof variant is read, the reference formal trace is partitioned into frozen semantic obligations, visualized in the paper as required tactic regions. An obligation O_k records required inputs, a principal composed Goal–Context Delta, produced output, predecessor-region relations, stable identifiers, and confidence. The set $\mathcal{O}^+ = \{O_1, \ldots, O_K\}$ and its hash are fixed for every variant in a theorem group. Removing an informal bridge therefore cannot also remove the denominator used to detect the omission.

A required region is covered only when the matched informal transition supports both its principal effect and its output. A conclusion-producing transition cannot absorb all earlier bridge regions merely because it agrees with final goal closure. Shared facts and connector transitions may support several alignments, but they do not erase an uncovered required region.

Proposal, checks, and outputs. The obligation-partition prompt receives the theorem, Formal Delta Cards, and formal dependencies without a natural-proof variant. It asks the LLM to return each obligation's member IDs, semantic core, internal connectors, closure actions, and predecessor obligations. Its format requires every supplied action to belong to one obligation and predecessor references to form an acyclic dependency structure. Identifier and structural validation precede freezing the resulting obligation records and hash. Subsequent alignment receives this fixed set and proposes a core-and-connector region for each informal transition; deterministic checks validate the cited IDs, connectivity, roles, and scope. Reverse matching then returns input, effect, output, and method scores for each obligation, which are recomputed with the fixed gate and weights in Appendix E. Appendix G gives the input/output contracts and prompts for these separate stages.

A one-obligation example. The trace in Appendix I.8 yields one frozen obligation, `O000`: preserve periodicity when a real-valued function is coerced to a complex-valued function. The `exact congrArg` action supplies the semantic core, while `intro x` is an internal connector. These are two actions realizing one required effect, rather than two separately scored obligations. The archived reverse match cites natural transition `N5`; the case study shows all four component scores, the coverage gate, and the resulting coverage value.

D.3 Route-preserving compression

Compression is accepted when several tactic transitions express one externally visible inference with the same inputs, outputs, and dependency boundary as an informal transition. It is not accepted merely because a tactic is opaque or closes the target. A theorem proved through another principal lemma, case structure, or construction remains a valid alternative route and is distinguishable from faithful compression.

The certificate records the informal transition, its connected tactic region, whether the relation is a same-route compression, and which frozen required regions support it. If these references cannot be verified against the supplied transition packet, the evaluator abstains on the example rather than matching it by a local heuristic.

E Scoring and Diagnostic Protocol

E.1 Informal-step alignment

For every informal transition r_i with nonzero importance, the alignment certificate identifies a connected tactic region $\mathcal{R}_i$ and cites endpoint-state, Goal–Context Delta, output, and premise evidence. The local score $m_i \in [0, 1]$ compares the conditions used in the source reasoning state with formal inputs, the informal effect with the composed state change, and the produced conclusion with the region output. It also checks theorem or rule use, branch scope, and witnesses when applicable. Importance α_i weights proof-level aggregation and prioritizes competing region assignments; it does not directly increase m_i for the selected region.

The local calibration is evidence constrained. A contradiction forces $m_i = 0$. A semantically related output without the principal Goal–Context Delta is capped at 0.45, and a matching principal Delta without the required hypotheses is capped at 0.70. Scores above 0.90 require direct output evidence together with the relevant inputs and state transition. Table 7 summarizes these unchanged anchors.

The informal-step alignment score is

$$M_N = \frac{\sum_{i:\alpha_i>0} \alpha_i m_i}{\sum_{i:\alpha_i>0} \alpha_i}. \tag{11}$$

Transitions with $\alpha_i = 0$ remain in the certificate for provenance but do not enter the denominator.

E.2 Tactic-region coverage

Tactic-region coverage asks whether the natural proof expresses the frozen required regions realized by the specified formal route. For informal transition r_i and required region O_k, the matcher returns input, effect, output, and method scores. Their weights and interpretations remain fixed as shown in

Table 7: Evidence anchors for informal-transition alignment.

Evidence condition	Score constraint
Contradicted transition	$m_i = 0$
Surface or thematic relation only	normally $m_i \leq 0.20$
Principal Delta absent	$m_i \leq 0.45$
Required hypotheses absent	$m_i \leq 0.70$
Strong but partly implicit match	$0.70 < m_i \leq 0.90$
Direct complete realization	$0.90 < m_i \leq 1$

Table 8: Fixed required-region match components.

Component	Weight	Comparison
Input	0.15	natural conditions versus required formal inputs
Effect	0.40	natural effect versus composed Goal–Context Delta
Output	0.35	produced natural conclusion versus formal output
Method	0.10	principal theorem, construction, or proof rule

Table 8; effect and output receive most of the mass because route coverage requires both the relevant state change and its result.

In supplementary testing on 34 additional ReasBook theorem groups outside the reported evaluation set, we selected $(0.15, 0.40, 0.35, 0.10)$ as a well-performing configuration based on ranking performance and grouped validation. These development groups are disjoint from the current evaluation population; semantic matches, outer aggregation coefficients, and coverage thresholds were held fixed.

Writing the component scores as s_{ik}^{in}, s_{ik}^{eff}, s_{ik}^{out}, and s_{ik}^{method}, let g_{ik} indicate that r_i is cited and that both effect and output scores are at least 0.5. The deterministic recomputation is unchanged:

$$s_{ik} = 0.15 s_{ik}^{\text{in}} + 0.40 s_{ik}^{\text{eff}} + 0.35 s_{ik}^{\text{out}} + 0.10 s_{ik}^{\text{method}}, \tag{12}$$

$$c_k = \max_i (g_{ik} s_{ik}), \tag{13}$$

$$M_F = \frac{1}{K} \sum_{k=1}^{K} c_k. \tag{14}$$

The gate prevents a final-statement or method-name match from covering an intermediate required region by itself. All K regions are included with equal weight, including uncovered regions with $c_k = 0$. Because the set is frozen before any natural variant is read, deleting a bridge cannot shrink the denominator.

E.3 Whole-proof and dependency terms

The final stage validates the proposed partition and returns a proof-specific coherence score Q. The validated partition and Q are then fixed for that proof. The score considers central claims, final support, missing bridges, case completeness, unsupported jumps, and contradictions; it is recorded separately rather than computed as an average of m_i or c_k.

Concretely, Stage D receives the natural proof, formal evidence, and the proposed alignment partition. It may return partition corrections and a `global_judgment` for the proof, including `global_score`, structural flags, and supporting evidence. Corrections are applied to the preceding partition before validation; the resulting `global_score` supplies Q. The full input/output protocol is given in Appendix G. In the example of Appendix I.8, candidate `P7` has $Q = 0.90$ and no flagged missing bridge, whereas its informal-step score is approximately $M_N = 0.504586$. These recorded outputs illustrate the distinction between whole-proof coherence and local support.

Table 9: Diagnostic categories retained by the alignment certificate.

Diagnostic	Interpretation
Missing informal support	A substantive transition has no validated tactic region.
Uncovered required region	A frozen formal requirement lacks natural coverage.
Dependency mismatch	Aligned transitions do not preserve direction or branch scope.
Formal extra route	A proof-relevant region lies outside the reference-supported route.
Cycle evidence	Strong natural dependencies form a directed cycle.

The dependency-direction score R compares D_N with directed reachability in D_F. Let $\rho_F(a, b)$ be the widest-path value from formal transition a to b, obtained by maximizing the minimum confidence along each directed path. For a natural dependency $d = (r_i, r_j) \in D_N$ with core-transition sets $\mathcal{C}_i$ and $\mathcal{C}_j$, define

$$f_d = \max_{a \in \mathcal{C}_i,\, b \in \mathcal{C}_j} \rho_F(a, b), \tag{15}$$

$$b_d = \max_{a \in \mathcal{C}_i,\, b \in \mathcal{C}_j} \rho_F(b, a), \tag{16}$$

$$r_d = \begin{cases} f_d/(f_d + b_d), & f_d + b_d > 0, \\ 1/2, & f_d + b_d = 0, \end{cases} \tag{17}$$

$$R = \frac{\sum_{d \in D_N} q_d r_d}{\sum_{d \in D_N} q_d}, \tag{18}$$

where q_d is natural-dependency confidence. If no scored dependency is available, $R = 1/2$. Thus missing direction evidence is neutral rather than interpreted as agreement or reversal. This term measures dependency-direction consistency among aligned regions; it does not compare graph size, path length, branching count, or graph isomorphism.

The structural-soundness factor is replayed from the strong natural dependencies,

$$C = 1 - \frac{\sum_{d \in D_N^{\mathrm{cyc}}} q_d}{\sum_{d \in D_N} q_d}, \tag{19}$$

with $C = 1$ when no scored strong dependency participates in a directed cycle. The reported score remains

$$S_{\mathrm{GCDA}} = C \cdot \frac{M_N + M_F + Q + R}{4}. \tag{20}$$

Alternative geometric, harmonic, minimum, and product aggregations in Appendix H use the same frozen component outputs and are not separate learned metrics or test-set formula searches.

E.4 Certificate validation and diagnostics

Before scoring, the validator requires every nonzero-importance informal transition and every frozen required region to appear exactly once in the certificate. All cited state, transition-record, fact, and region identifiers must occur in the input packet. Core and connector transitions must form a connected region, connectors receive no independent semantic credit, branch assignments must be scope compatible, and claimed dependency paths must follow D_F. One repair may correct schema or identifier errors using the validator's error list; a second failure yields abstention.

F Corpus Provenance and Controlled Variants

F.1 Source corpora

The ReasBook-derived corpus is constructed from Lean 4 formalization artifacts associated with textbook- and paper-level mathematical developments ReasBook Contributors (2026); Wang et al. (2026b). We retain theorem groups for which a natural proof description, a target theorem, and a verified Lean proof are available. The anonymized supplementary artifact records theorem-group identifiers, source locators when available, formal declarations, validation status, and content hashes.

Table 10: Exact ReasBook-derived source counts recovered from the frozen provenance manifest.

Source family	Theorem groups
Analysis II (Tao)	39
Basic Analysis I (Lebl)	28
Convex Analysis (Rockafellar)	46
Smooth Minimization of Non-Smooth Functions (Nesterov)	16
Foundational Algebra, Chapter 1 corpus	36
Combinatorial Group Theory (Magnus et al.)	23
Linear Representations of Finite Groups (Serre)	31
The Stacks Project	47
Probability Theory: A Comprehensive Course (Klenke)	39
A Concise Course in Algebraic Topology (May)	40
Introduction to Smooth Manifolds (Lee)	44
Elementary Theory of Analytic Functions (Cartan)	43
Total	**432**

Table 11: Evaluation corpus sizes.

Corpus	Groups	Pos.	Neg.	Total	Types
ReasBook	432	1,001	2,104	3,105	5
miniF2F	238	656	1,127	1,783	5
Total	**670**	**1,657**	**3,231**	**4,888**	–

The corpus contains 432 theorem groups and 3,105 variants across real and functional analysis, convex analysis and optimization, algebra, algebraic geometry, probability, topology and differential geometry, and complex analysis.

The miniF2F corpus contains problems from AMC, AIME, IMO, MATH, and related olympiad-style collections Zheng et al. (2022). We begin from 244 candidate Lean 4 records and retain 238 whose collected formal proofs compile in the frozen environment. Its final five-type population contains 1,783 variants.

The frozen ReasBook provenance manifest contains 432 records and resolves all groups to twelve source families. Table 10 reports the exact counts from that manifest, with source titles checked against the cited publications.

Frozen evaluation population. All reported evaluations use the frozen ProofRouteBench manifest, comprising 432 ReasBook groups and 238 miniF2F groups (670 groups and 4,888 variants in total). The same theorem-group population and eligible variants are used for every method and main-table aggregate. Component analyses use the corresponding stored outputs, and supplementary analyses specify their sample sizes and input requirements. The reported results characterize this frozen evaluation population.

F.2 Variant protocol

Each theorem group fixes the problem statement, target theorem, verified formal proof, and frozen obligation set. Process-preserving controls are the original proof, a faithful paraphrase, and nonrequired-step removal. The five officially reported process changes are required-bridge removal, invalid rewrite, wrong dependency, circular dependency, and alternative strategy.

Required-bridge removal deletes an intermediate unit on an ancestor path to the conclusion. Invalid rewrite changes a local formula, symbol, or conclusion while retaining the local position. Wrong dependency redirects or reverses a semantic edge while preserving node content when possible; a deterministic audit separately records whether the edit also creates a cycle. Circular dependency introduces a descendant-to-ancestor relation. These two dependency variants operate on the explicit dependency certificate D_N of the recovered natural graph, so the surface proof text can remain unchanged. They therefore test the evaluator's use of supplied structural evidence rather than its ability

Table 12: Variant counts in the supplementary evaluation population.

Variant	ReasBook	miniF2F	Primary construct
Original	432	238	Reference route
Faithful paraphrase	432	238	Surface stability
Nonrequired removal	137	180	Noncritical granularity
Bridge removed	378	183	Required-region coverage
Invalid rewrite	432	238	Local semantics
Wrong dependency	431	234	Dependency direction
Circular dependency	431	234	Acyclicity
Alternative strategy	432	238	Correct proof, different route
Total	**3,105**	**1,783**	

to recover a hidden dependency from raw prose. Alternative strategy preserves the mathematical correctness of the proof but changes a principal lemma, construction, branch structure, or governing proof principle. It is a route-different negative even when the resulting formal theorem remains valid.

Variant counts differ because each transformation is applied only when its structural preconditions can be verified. For example, removals require an eligible noncritical step or necessary bridge, while dependency edits require a modifiable validated edge; theorem groups without the corresponding structure simply omit that variant type.

Each variant receives a binary label indicating whether it preserves the reference process or realizes a controlled process change. Before scoring, variant names, mutation targets, labels, previous scores, and selection rationales are removed. GCDA then scores each anonymized variant against the corresponding formal trace, and the external baselines score each proof pair independently. Rows that fail formal or structural validation are excluded from the official aggregates rather than assigned a default low score.

G Implementation and Reproducibility Protocol

G.1 Information boundary and theorem-group batching

All methods use the same theorem-group manifest and are isolated from the variant name, mutation description, gold label, expected ranking, and prior score. Endpoint-only methods receive theorem statements. Text and embedding methods receive the natural and rendered formal proof texts, and the external proof-level judges score each proof pair independently. GCDA may organize computation by theorem group to reuse a fixed formal trace, but its final validation and frozen-obligation matching stages process exactly one natural proof per scoring request. Each request contains only the target proof, its own intermediate records, and the fixed formal evidence, and returns a certificate and score for that proof ID. Independently computed scores may subsequently be compared within theorem groups.

Frozen required tactic regions are constructed from the reference formal trace before any natural variant is scored. The scorer verifies theorem-level identity of the obligation hash and validates every state, transition, and region identifier against the input packet. A certificate that remains invalid after the single repair attempt is recorded as an abstention; it is not replaced by a local text match.

G.2 Model and inference configuration

All language-model-dependent GCDA scoring stages use GPT-5.4 with the fixed configuration in Table 13. The context budget determines only whether alignment is evaluated in one global request or by coarse routing followed by local partitioning. Both modes retain the complete natural proof as context for final scoring. Alternative-strategy examples used in benchmark construction were generated separately with GPT-5.6-sol and then subjected to theorem-validity and route-change validation; that model is not the GCDA scorer.

Table 13: Frozen inference configuration for GCDA scoring.

Setting	Final value
Scoring model	GPT-5.4
Temperature	0
Reasoning effort	medium
Context window	400,000 tokens
Safe input budget	340,000 tokens
Maximum output	6,000 tokens
Request timeout	180 seconds
Streaming	enabled
Maximum request concurrency	60
Structured repair attempts	at most one

For requests that fit the safe budget, Stage C performs one joint partition over the complete natural and formal graphs. Larger packets use a score-free coarse-routing request and then evaluate local formal neighborhoods in detailed partition requests. The deterministic merge resolves overlapping candidates and validates the resulting global certificate. Routing changes the amount of formal evidence shown in each detailed request, not the scoring definition.

G.3 Baseline implementations

Common comparison protocol. Each baseline receives the same natural-proof variant and verified formal proof, represented according to the method's declared input boundary, and returns one scalar for which a larger value denotes stronger correspondence. For each negative type, this scalar is used to rank process-preserving and negative variants only within the same theorem group. Variant names, labels, mutation descriptions, and expected rankings are excluded from every request, so the comparison tests the metric rather than its access to construction metadata.

Lexical overlap. The common renderer converts each informal transition into a premise–claim description and each formal transition into an event description containing its action, before and after goals, and observable used or introduced facts. Concatenating these descriptions gives the two proof documents used by BLEU, chrF, and ROUGE Papineni et al. (2002); Popović (2015); Lin (2004). Sentence-level BLEU and chrF are computed in both directions and averaged; ROUGE-1, ROUGE-2, and ROUGE-L use their F-measures with stemming. The resulting document score is applied directly to each benchmark row, without step alignment or dependency information.

Neural semantic similarity. E5 step F1 and BERTScore first compare the rendered informal and formal step lists Wang et al. (2022); Zhang et al. (2020). E5 embeds every step with `intfloat/e5-base-v2`; pairwise cosine similarities are mapped to $[0, 1]$, each step takes its best match in the opposite proof, and the two directional means are combined by their harmonic mean. BERTScore applies the same bidirectional maximum-matching reduction to all pairs of 384-token chunks using `roberta-large`. Symmetric BARTScore computes conditional likelihood in both directions for each chunk pair, combines the two directions geometrically, and then uses the same maximum-matching reduction Yuan et al. (2021). The ROSCOE-style score embeds steps with `all-MiniLM-L6-v2` and uses reasoning alignment, the mean best formal support for the informal steps, as its ranking score Golovneva et al. (2023). MoverScore instead compares the two concatenated proof documents with unigram transport and uniform inverse-document-frequency weights Zhao et al. (2019).

Proof-level adaptations of statement evaluation and retrieval. LeanSearch-proof adapts the E5-Mistral retrieval encoder used in LeanSearch Gao et al. (2024) to encode the theorem together with the complete proof on each side using `intfloat/e5-mistral-7b-instruct`; the affine-mapped cosine between the normalized natural and formal embeddings is the proof-pair score. LeanScorer-proof asks the judge to decompose the informal argument into substantive steps and assign each step an A, B, or C consistency label against the Lean proof, followed by mandatory coverage and dependency labels Yu et al. (2026). Our adaptation labels the three consistency levels A/B/C, maps them to 1, 0.5, and 0, and aggregates them by a fixed Sugeno rule with an explicit

major-inconsistency veto. These are out-of-distribution proof-level adaptations, not reproductions of the original statement-level tasks.

Holistic LLM judgments. Direct scalar reads the complete proof pair rendered as informal steps and formal events and returns one overall correspondence score. Direct rubric uses the same evidence and fixed score anchors, but also identifies missing formal obligations and unsupported informal steps; its returned overall score remains the ranking value. Our ProofScore adaptation fixes formal validity to one and combines elicited coverage and dependency scores by a geometric mean with a critical-error cap, rather than reproducing the original node-wise formula Cabral et al. (2026). All three methods judge each row independently and receive no cross-variant information.

Structural controls and ablations. Holistic components asks the same model to predict the named GCDA components and an overall score without producing node–region certificates. Holistic formula recomputes the fixed component formula from that response, whereas One-way structured requires support for every informal step but omits frozen formal-to-natural obligation coverage. These controls examine the contribution of explicit evidence certificates and bidirectional coverage. The remaining formula and component analyses in Appendix H replay available stored GCDA component values without fresh model calls; they change the final aggregation or omit one recorded component.

Statement-only controls. Endpoint metrics compare only the natural and Lean theorem statements, ignoring both proof bodies. All variants in a theorem group share these endpoints, so each control necessarily assigns the same score throughout that group. Under the tie-aware theorem-equal AUC used in the benchmark, this produces the expected value of 0.5 and serves as a check that statement agreement alone cannot distinguish process changes.

G.4 Run counts, randomness, and grouped statistics

The primary result contains one complete GCDA score for every eligible row in both corpora. The GCDA stability analysis in Appendix H compares paired outputs from independent executions of every language-model-dependent stage on 165 ReasBook groups in the current evaluation population. LLM baselines with repeated measurements use two independent calls under the same configuration and report their arithmetic mean; deterministic lexical and embedding metrics are executed once per input. These repetitions estimate stochastic variation and are not treated as separate methods.

Language-model requests use temperature zero, but hosted inference is not assumed to be bitwise deterministic. Reproducibility therefore relies on frozen prompts, model and inference settings, exact input and response hashes, parser and validator records, and the independent-repeat analysis. Deterministic routing uses the string seed `iw-dpa-full-benchmark-gpt54-v1`. The theorem-group cluster bootstrap uses 2,000 replicates with base seed 20260721; derived per-scope seeds are recorded in the result artifact.

For each negative type, positive–negative comparisons are formed only within the same theorem group. Pooled AUC uses all eligible rows, within-theorem AUC weights same-theorem pairs, and theorem-equal AUC averages valid theorem-level AUCs equally. MacroAUC and WorstAUC summarize the five per-type theorem-equal values. The bootstrap resamples theorem groups rather than individual rows, preserving dependence among variants from the same theorem.

G.5 Software and computing infrastructure

Formal extraction uses Lean 4.31.0 and mathlib revision `fabf563a7c95a166b8d7b6efca11c8b4dc9d911f`. The supplementary Python package requires Python 3.10 or later and pins `leanclient` 0.11.0 for Lean communication. Batch preprocessing, validation, and aggregation run on Linux CPU nodes; the largest retained job configuration uses 64 CPU workers and 64 GB memory. The exact CPU SKU was not retained, and these deterministic stages do not use architecture-specific kernels.

The GPT-based stages use hosted inference, so no local GPU is used for GCDA scoring. The E5-Mistral embedding baseline uses `intfloat/e5-mistral-7b-instruct`, model snapshot `d62cc4417be0e3289c03dd459d1ef2d137e2554e`, served by vLLM 0.11.0 in bfloat16 on one NVIDIA H100 80 GB GPU with a 4,096-token model length and deterministic server seed 0.

The supplementary environment file records the remaining package versions and third-party model revisions used by the baselines.

G.6 Prompt assembly and output constraints

The following subsections reproduce the frozen system messages for the principal GCDA stages and the three LLM baselines used in the main comparison. At run time, each system message is paired with a machine-generated JSON user message containing the theorem, the permitted evidence packet, the required output schema, and the prompt version. The user-message constructors and schemas are included in the supplementary code; variant metadata and labels are rejected before dispatch. Listings preserve the original prompt text.

G.7 Natural-transition analysis prompt

Input: theorem statement and one already-extracted natural graph (node IDs, text, edges, and conclusion IDs). Output: one analysis per natural ID, including logical type, importance, removal counterfactual, and expected Delta. This stage annotates existing nodes; raw-text segmentation precedes it.

Abbreviated output field map (the request carries the full JSON schema):

```
natural_node_analyses: [{natural_id, logical_type, importance_score,
  has_logical_function, counterfactual_if_removed, expected_delta}]
```

This prompt jointly assigns logical roles, importance, removal counterfactuals, and expected semantic state changes to all natural transitions.

```
You are a mathematical proof-structure analyst.

Analyze all natural-language proof nodes jointly.

For every natural node:

1. Classify its logical type.
2. Assign a continuous logical-importance score in [0,1].
3. Rank all nodes by importance within the proof.
4. State whether the node has any logical function.
5. Explain what would happen if the node were removed.
6. Infer the semantic proof-state transition that the node is expected
   to perform.

Importance measures logical contribution, not sentence length,
formula density, writing style, or rhetorical emphasis.

Use this calibration:

- exactly 0:
  The sentence has no logical function. It is explanatory, repetitive,
  rhetorical, transitional, or purely descriptive. Removing it changes
  no available hypothesis, inference, branch, goal, or conclusion.
  A sentence containing formulas may still receive score 0 if it
      performs
  no logical role.

- 0.01 to 0.10:
  Almost no independent logical contribution. It only restates or
      lightly
  expands a nearby step.

- 0.10 to 0.30:
  Low-impact but genuine proof step.

- 0.30 to 0.60:
  Substantive derivation, calculation, rewrite, or local goal
      reduction.
```

```
36
37  - 0.60 to 0.85:
38    Key intermediate lemma, important witness, major goal reduction,
39    or proof-controlling case split.
40
41  - 0.85 to 1.00:
42    Final conclusion, indispensable bridge, or a step without which the
43    proof cannot reach its conclusion.
44
45  Evaluate nodes comparatively. The numeric scores, importance ranking,
46  importance bands, and removal counterfactuals must be mutually
        consistent.
47
48  Allowed logical types:
49
50  - assumption
51  - definition
52  - derived_fact
53  - rewrite
54  - goal_reduction
55  - case_split
56  - witness
57  - conclusion
58  - expository
59
60  For expected_delta, describe semantic effects rather than Lean syntax:
61
62  - input claims
63  - used hypotheses
64  - before-goal semantics
65  - after-goal semantics
66  - goal operation
67  - rewritten goals
68  - new goals
69  - closed goals
70  - hypotheses introduced
71  - hypotheses removed
72  - hypotheses changed
73  - branch or scope change
74  - witness
75  - output claim
76  - theorem or rule use
77
78  Do not use evaluation labels, expected quality ranks, or variant names
        .
79  Return strict JSON only.
```

G.8 FROZEN-OBLIGATION PARTITION PROMPT

Input: theorem statement, allowed formal IDs, formal action cards, and formal edges; no natural proof is supplied. Output: required regions with inputs, outputs, member/core/connector IDs, and predecessor obligations. Validation canonicalizes IDs and freezes the resulting set.

Abbreviated output field map (the request carries the full JSON schema):

```
obligations: [{obligation_id, semantic_goal, required_inputs,
  produced_claims, member_formal_ids, core_formal_ids,
  internal_connector_ids, predecessor_obligation_ids}]
```

This prompt groups the verified formal trajectory into a compact directed acyclic graph of theorem-level required regions.

```
1  You build a semantic obligation graph from one formal proof trajectory
       .
2
```

```
You receive only the theorem statement and a graph of positive formal delta cards. Group the low-level formal actions into a small ordered set of externally meaningful mathematical obligations.

Rules:
- Every allowed formal ID must appear in exactly one member_formal_ids list.
- Each obligation must express one coherent mathematical effect.
- Named or reusable intermediate claims normally form obligation boundaries.
- Do not let a final automation or closure action absorb independent internal bridges.
- core_formal_ids carry the mathematical effect.
- internal_connector_ids only connect or transport information.
- macro_closure_ids close an already established obligation.
- The three role lists must be disjoint subsets of member_formal_ids.
- Every obligation needs at least one core formal ID.
- predecessor_obligation_ids must form a DAG and use only IDs you define.
- Prefer a compact semantic graph, but never merge distinct causal branches or independent bridge claims merely to reduce the count.

Return strict JSON only. Do not invent formal IDs.
```

G.9 GLOBAL PARTITION PROMPT

Input: one natural graph and its Stage A analyses, together with formal Delta cards and edges. Output: natural regions, formal-node assignments, unmatched nodes, and partition conflicts. Each region cites core, connector, and shared-evidence IDs and returns local scores and evidence.

Abbreviated output field map (the request carries the full JSON schema):

```
natural_regions: [{natural_id, core_formal_ids, connector_formal_ids,
  shared_evidence_ids, composed_region_delta, match_score, evidence}]
formal_node_assignments: [...]
unmatched_formal_nodes: [...]
partition_conflicts: [...]
```

This prompt is used when the complete natural and formal packets fit in one request.

```
You are a formal proof-graph partition and natural-to-formal alignment judge.

You are given:

1. A theorem statement.
2. A complete natural-language proof graph.
3. A joint analysis of all natural nodes, including:
   - logical type
   - continuous importance in [0,1]
   - expected semantic delta
   - removal counterfactual
4. A complete formal proof graph.
5. A delta card for every formal node.
6. Formal causal edges and source order.

Partition the formal graph jointly according to the natural proof steps.

Every formal node must receive one of these roles:

1. core:
   The formal node realizes an essential part of a natural node's primary
   goal/context delta. Core nodes receive semantic content credit.

```

```
2. connector:
   The formal node only connects core nodes or preserves goal lineage.
   Connector nodes receive no semantic content credit.

3. shared_evidence:
   The node or attached fact provides a reusable assumption, theorem,
   hypothesis, or context item used by multiple natural nodes.
   Shared evidence may support hypothesis-use matching but must not be
   counted as multiple primary deltas.

4. unmatched_or_administrative:
   The node is administrative, omitted by the natural proof,
      technically
   necessary but not explicitly represented, or has no reliable
      natural
   counterpart.

The formal goal/context delta is the primary evidence:

- before_targets
- after_targets
- rewritten_goals
- new_goals
- solved_goals
- closed_goals
- remaining_goals
- unchanged_goals
- before_context
- after_context
- hypothesis_delta.introduced
- hypothesis_delta.removed
- hypothesis_delta.changed
- hypothesis_delta.used_candidates

Tactic text, action kind, used facts, theorem names, and constants are
supporting evidence. They cannot replace an actual matching delta.

A natural node may correspond to multiple formal nodes.

Continuous natural-node importance controls allocation priority when
multiple natural nodes compete for the same non-shareable core
    transition.
Importance must not directly increase match scores.

A high-importance node with weak delta correspondence must still
    receive
a low match score.

Natural nodes with importance exactly 0 do not require a formal region
    ,
but they must remain visible in the output.

For every nonzero-importance natural node:

1. Select core formal IDs.
2. Select connector formal IDs.
3. Select shared-evidence IDs.
4. Describe the composed region delta.
5. Compare expected natural delta against formal region delta.
6. Output continuous component scores and an overall match score in
    [0,1].
7. Cite exact formal IDs and exact delta fields.
8. List missing expected effects.
9. Mark contradictions.
```

```
Match-score calibration:

- 0:
  unrelated or contradicted; the formal region does not realize the
  natural logical transition

- 0.01 to 0.20:
  surface or thematic relation only

- 0.20 to 0.45:
  partial relation, but the primary delta is missing

- 0.45 to 0.70:
  part of the intended transition is realized, with substantial
      omissions

- 0.70 to 0.90:
  strong delta match with limited implicitness or granularity mismatch

- 0.90 to 1.00:
  complete and direct realization of the natural logical transition

If contradiction is true, match_score must be 0.

If the output claim is similar but the required goal/context delta is
absent, match_score must not exceed 0.45.

If the primary delta matches but required hypotheses are missing,
match_score must not exceed 0.70.

Use only supplied IDs.
Do not use evaluation labels, variant names, or expected ranks.
Return strict JSON only.
```

G.10 COARSE-ROUTING PROMPT

Input: Stage A analyses, compact formal Delta signatures, and the allowed IDs. Output: anchors, source-order intervals, lineage IDs, and neighborhood-expansion hints for each substantive natural node; this request returns no alignment score.

Abbreviated output field map (the request carries the full JSON schema):

```
routing: [{natural_id, anchor_formal_ids, source_order_intervals,
  goal_lineage_ids, shared_evidence_ids, expand_predecessor_hops,
  expand_successor_hops, routing_confidence, reason}]
```

This score-free prompt identifies formal neighborhoods for large proof packets.

```
You are a coarse proof-graph routing model.

You are given all natural proof nodes and compact signatures of all
formal nodes.

For every nonzero-importance natural node, identify:

- likely anchor formal IDs
- likely source-order intervals
- likely goal-lineage IDs
- likely shared hypotheses or theorem evidence
- a small set of formal neighborhoods that should be expanded

This is routing only.

Do not assign final match scores.
Do not produce final partitions.
Do not require one natural node to map to one formal node.
```

```

Prefer recall over precision, but keep each natural node's routed formal
set compact enough for a detailed follow-up request.

Use expected natural delta and formal delta signatures.
Do not rely primarily on tactic names.
Return strict JSON only.
```

G.11 LOCAL PARTITION PROMPT

Input: a batch of natural nodes and their analyses, routed formal cards and internal edges, and explicit allowed/required IDs. Output: the global-partition fields for that batch, plus cross-batch claims; deterministic merging constructs the complete proposal.

Abbreviated output field map (the request carries the full JSON schema):

```
natural_regions: [...]
formal_node_assignments: [...]
unmatched_formal_nodes: [...]
partition_conflicts: [...]
cross_batch_claims: [...]
```

This prompt evaluates the routed neighborhoods and marks exclusive and shareable candidates for deterministic global merging.

```
You are a formal proof-graph partition and natural-to-formal alignment judge.

You are given:

1. A theorem statement.
2. A complete natural-language proof graph.
3. A joint analysis of all natural nodes, including:
   - logical type
   - continuous importance in [0,1]
   - expected semantic delta
   - removal counterfactual
4. A complete formal proof graph.
5. A delta card for every formal node.
6. Formal causal edges and source order.

Partition the formal graph jointly according to the natural proof steps.

Every formal node must receive one of these roles:

1. core:
   The formal node realizes an essential part of a natural node's primary
   goal/context delta. Core nodes receive semantic content credit.

2. connector:
   The formal node only connects core nodes or preserves goal lineage.
   Connector nodes receive no semantic content credit.

3. shared_evidence:
   The node or attached fact provides a reusable assumption, theorem,
   hypothesis, or context item used by multiple natural nodes.
   Shared evidence may support hypothesis-use matching but must not be
   counted as multiple primary deltas.

4. unmatched_or_administrative:
   The node is administrative, omitted by the natural proof, technically
```

```
    necessary but not explicitly represented, or has no reliable natural
    counterpart.

The formal goal/context delta is the primary evidence:

- before_targets
- after_targets
- rewritten_goals
- new_goals
- solved_goals
- closed_goals
- remaining_goals
- unchanged_goals
- before_context
- after_context
- hypothesis_delta.introduced
- hypothesis_delta.removed
- hypothesis_delta.changed
- hypothesis_delta.used_candidates

Tactic text, action kind, used facts, theorem names, and constants are
supporting evidence. They cannot replace an actual matching delta.

A natural node may correspond to multiple formal nodes.

Continuous natural-node importance controls allocation priority when
multiple natural nodes compete for the same non-shareable core transition.
Importance must not directly increase match scores.

A high-importance node with weak delta correspondence must still receive
a low match score.

Natural nodes with importance exactly 0 do not require a formal region,
but they must remain visible in the output.

For every nonzero-importance natural node:

1. Select core formal IDs.
2. Select connector formal IDs.
3. Select shared-evidence IDs.
4. Describe the composed region delta.
5. Compare expected natural delta against formal region delta.
6. Output continuous component scores and an overall match score in [0,1].
7. Cite exact formal IDs and exact delta fields.
8. List missing expected effects.
9. Mark contradictions.

Match-score calibration:

- 0:
  unrelated or contradicted; the formal region does not realize the
  natural logical transition

- 0.01 to 0.20:
  surface or thematic relation only

- 0.20 to 0.45:
  partial relation, but the primary delta is missing

- 0.45 to 0.70:
```

```
  part of the intended transition is realized, with substantial
      omissions

- 0.70 to 0.90:
  strong delta match with limited implicitness or granularity mismatch

- 0.90 to 1.00:
  complete and direct realization of the natural logical transition

If contradiction is true, match_score must be 0.

If the output claim is similar but the required goal/context delta is
absent, match_score must not exceed 0.45.

If the primary delta matches but required hypotheses are missing,
match_score must not exceed 0.70.

Use only supplied IDs.
Do not use evaluation labels, variant names, or expected ranks.
Return strict JSON only.

This is a local detailed partition request produced by a previous
    coarse
routing stage.

Evaluate all natural nodes in this batch jointly.

The supplied formal subgraph may overlap with subgraphs used in other
batches. Do not assume that every formal node is exclusively available
    .

Mark every selected core node as:

- exclusive_core_candidate
or
- shareable_core_candidate

The final global merge stage will resolve cross-batch conflicts.

Return strict JSON only.
```

G.12 REQUIRED-REGION MATCHING PROMPT

Input: the theorem, frozen obligations, fixed formal evidence, and one target natural proof with its graph. Output: a JSON object with a `proof_results` array containing one result for the target proof. The result includes `proof_id` and `obligation_matches`, with one match per frozen obligation. Each match contains the obligation ID, a natural-node ID (or null), four component scores for input, effect, output, and method, the matched and missing effects, and supporting evidence, as shown below. Coverage credit is recomputed by the fixed scoring rule.

Abbreviated output field map (the request carries the full JSON schema):

```
proof_results: [{proof_id, obligation_matches: [{obligation_id,
  natural_id, input_score, effect_score, output_score, method_score,
  matched_effects, missing_effects, evidence}]}]
```

This reverse-direction prompt tests whether each frozen formal region is expressed by a natural transition.

```
You align the nodes of one target natural-language proof to a frozen
semantic-obligation graph.

Use only the target proof, its graph, and the supplied fixed formal
evidence.

For every obligation, select the single best node from the target
```

```
proof or null. Judge four components independently in [0,1]:
- input_score: required inputs and premises;
- effect_score: expected natural delta versus composed formal effect;
- output_score: natural claim versus produced obligation claims;
- method_score: proof-method compatibility, used only as a weak
  auxiliary signal.

An obligation is genuinely covered only when effect_score and
output_score are both at least 0.5. A conclusion node must not
receive credit for internal obligations merely because it also
closes the theorem. Use obligation-specific evidence.

Return exactly one proof_results entry for the target proof and
exactly one match row for every supplied obligation ID. Use only
that proof's supplied natural IDs and the supplied obligation IDs.
Return strict JSON only.
```

G.13 WHOLE-PROOF VALIDATION PROMPT

Input: one target proof with its proposed partition and the fixed formal trace. Output: a global judgment and structured partition corrections for that proof ID. In corrections-only mode, unchanged regions are inherited from the target proof's proposal before deterministic validation.

Abbreviated output field map (the request carries the full JSON schema):

```
proof_results: [{proof_id, validated_partition: {corrections: [...]},
  global_judgment: {global_score, proof_validity, conclusion_validity,
    dependency_preservation, missing_critical_bridge, evidence}}]
```

```
You are the final proof-alignment verifier.

You are given:

1. The theorem statement.
2. The complete natural proof graph.
3. Continuous natural-node importance scores.
4. The proposed natural-to-formal partition.
5. The composed formal delta for every selected region.
6. Natural dependency edges.
7. Formal causal edges.
8. Partition conflicts, unmatched important nodes, and unmatched
    formal nodes.
9. Node-region match scores.

First verify and, if necessary, minimally correct the partition:

- resolve conflicting exclusive core assignments
- preserve shared evidence when justified
- ensure connector nodes receive no semantic content credit
- ensure important natural nodes are not silently omitted
- ensure region order is compatible with formal causal structure

Then judge the global proof:

- whether important natural steps are formally supported
- whether natural dependencies follow the formal proof
- whether any key bridge is missing
- whether case splits are complete
- whether witnesses are valid
- whether the final conclusion is supported
- whether there are unsupported jumps
- whether there are mathematical contradictions

Route-faithfulness calibration:

```

```
- Separate mathematical correctness from alignment with the supplied formal
  proof route. A shared final conclusion alone does not establish route
  alignment.
- Respect formal compression. A solver or macro tactic may support routine
  algebraic rewrites and intermediate consequences without recording one
  formal node per natural step. Do not penalize paraphrase, harmless elision,
  variable renaming, reordered exposition, or added detail when the decisive
  mathematical route and dependency backbone remain the same.
- For each proof, identify a compact route signature: the central construction
  or witness, decisive theorem or transformation, essential intermediate
  claims, case structure, and dependency backbone. Compare signatures rather
  than surface wording or proof length.
- Determine the reference route only from the supplied formal trace,
  including its observable inputs, effects, outputs, and dependencies.
  If a solver macro hides an internal route, record the limitation in
  evidence. An unobserved internal route alone is not proof of a mismatch.
- Treat the target proof as materially alternative only when the
  supplied formal evidence establishes a difference in a decisive
  construction, theorem, case split, invariant, or dependency chain.
  Mere notation changes or expanded calculations are not a different
  route.
- Allow routine compression when observable inputs, effects, outputs,
  and dependency boundaries support it. Correct an under-credited
  region only when specific supplied formal evidence justifies the
  correction.
- Determine the target proof's global score using the unchanged
  Global-score calibration below and its own formal support, missing
  bridges, dependencies, contradictions, and conclusion. Ground every
  partition correction in evidence for the target proof.

Global-score calibration:

- 0:
  contradicted or mathematically invalid proof

- 0.01 to 0.30:
  major unsupported claims, wrong conclusion, or broken proof route

- 0.30 to 0.60:
  partially supported but incomplete or containing substantial gaps

- 0.60 to 0.85:
  mostly valid and supported, with limited implicitness or missing detail

- 0.85 to 1.00:
  valid proof whose important claims, dependencies, and conclusion are
  strongly supported by formal evidence

Do not simply average node match scores.
The global score must judge proof-level validity and dependency structure.

Evaluate only the single target proof in this request. Return exactly
one proof_results entry for its proof_id.
```

```

Do not use evaluation labels, variant names, or expected quality ranks
    .
Return strict JSON only.
```

G.14 CERTIFICATE-REPAIR PROMPT

Input: the target proof's original request, the invalid response, the required schema, and exact validator errors. Output: one corrected JSON object in the schema of the failed stage, preserving the supplied evidence boundary.

The repair prompt receives the invalid response and the validator's exact error list. It may correct schema and identifier errors once, without introducing new evidence.

```
You are repairing a structured proof-alignment JSON certificate.

The previous response failed validation.

You are given:

1. The original mathematical request for the single target proof.
2. The invalid response.
3. Exact validation errors.
4. Required natural IDs.
5. Allowed formal IDs.
6. The exact required JSON schema.

Repair only the identified errors.

Do not drop natural nodes.
Do not invent formal IDs.
Do not change valid evidence without necessity.
Do not use evaluation labels or expected ranks.

Use only the same target proof and fixed formal evidence as the
original request.
Return strict JSON only.
```

G.15 DIRECT-SCALAR BASELINE PROMPT

Input: one problem statement and natural/formal proof pair or trace. Output: an integer score from 0 to 100, confidence, and a short reason.

Abbreviated output field map (the request carries the full JSON schema):

```
faithfulness_score: integer in [0,100]
confidence: number in [0,1]
reason_short: string
```

The strongest direct baseline returns one scalar without a structured certificate.

```
You are judging whether a natural-language proof faithfully explains a
     Lean formal proof of the same theorem.

Use only the provided problem statement, natural proof, and formal
    proof or trace. Do not assume hidden labels or metadata.

Return a JSON object with exactly these fields:
{
  "faithfulness_score": 0,
  "confidence": 0.0,
  "reason_short": ""
}

The faithfulness_score is an integer from 0 to 100. Reward correct
    proof-process alignment, necessary intermediate steps, and
    dependencies. Penalize any unsupported or incomplete explanation.
```

G.16 DIRECT-RUBRIC BASELINE PROMPT

Input: the same pairwise evidence as the direct-scalar baseline. Output: its score/confidence/reason fields, plus Boolean missing-step and wrong-dependency flags.

Abbreviated output field map (the request carries the full JSON schema):

```
faithfulness_score: integer in [0,100]
missing_key_step: Boolean
wrong_dependency: Boolean
confidence: number in [0,1]
reason_short: string
```

The rubric baseline adds explicit missing-step and dependency judgments while retaining a holistic scalar output.

```
You are judging whether a natural-language proof faithfully explains a
     Lean formal proof of the same theorem.

Use only the provided problem statement, natural proof, and formal
    proof or trace. Do not assume hidden labels or metadata.

Return a JSON object with exactly these fields:
{
  "faithfulness_score": 0,
  "missing_key_step": false,
  "wrong_dependency": false,
  "confidence": 0.0,
  "reason_short": ""
}

The faithfulness_score is an integer from 0 to 100. Reward correct
    proof-process alignment, necessary intermediate steps, and
    dependencies. Penalize missing central steps, reversed dependency,
     circular use of the conclusion, or unsupported claims.
```

G.17 LEANSCORER-PROOF ADAPTATION PROMPT

Input: one natural/formal proof pair or trace. Output: substantive natural steps with A/B/C labels, Lean evidence and reasons, plus coverage and dependency checks with the same label inventory.

Abbreviated output field map (the request carries the full JSON schema):

```
steps: [{step_id, natural_step, label, lean_evidence, reason}]
coverage_check: {label, reason}
dependency_check: {label, reason}
overall_reason: string
```

This prompt adapts LeanScorer's graded consistency labels to substantive steps in complete proof pairs.

```
You are evaluating whether a natural-language proof faithfully
    explains a Lean formal proof of the same theorem.

This is a proof-level adaptation of LeanScorer. Decompose the natural
    proof into its substantive mathematical steps and compare every
    step against the Lean proof or trace. Judge proof-process
    faithfulness, not merely whether both texts end at the same
    theorem.

Use exactly these labels:
- A: Perfectly Match. The step is mathematically correct, relevant,
    and supported by the Lean proof, possibly with harmless
    differences in notation or compression.
- B: Minor Inconsistency. The step is basically valid but has a small
    omission, harmless imprecision, or compression that does not
    change the proof strategy.
- C: Major Inconsistency. The step is materially unsupported or wrong,
     a required bridge is absent, the reasoning is circular, a later
```

```
      conclusion is used as a premise, or the claimed dependency order
      conflicts with the Lean proof.

In addition to the step labels, perform two mandatory checks:
- coverage_check: A if no necessary proof bridge is missing; B for a
      minor implicit step; C when a necessary intermediate justification
       is missing.
- dependency_check: A for a valid reasoning order; B if the order is
      unclear but not invalid; C for circular reasoning, reverse
      dependency, or use of a future/concluding claim as a premise.

Do not infer mutation names or hidden labels. Do not reward stylistic
      similarity. Do not treat a short automation tactic as evidence
      that every natural-language claim is supported; interpret the
      theorem and tactic semantics as carefully as possible.

Return one JSON object with exactly this shape:
{
  "steps": [
    {
      "step_id": "S1",
      "natural_step": "short verbatim or faithful restatement",
      "label": "A",
      "lean_evidence": "relevant Lean code or semantic explanation",
      "reason": "short reason"
    }
  ],
  "coverage_check": {
    "label": "A",
    "reason": "short reason"
  },
  "dependency_check": {
    "label": "A",
    "reason": "short reason"
  },
  "overall_reason": "one concise summary"
}

Return JSON only. Produce at least one step. Labels must be exactly A,
     B, or C.
```

H Complete Results and Audits

H.1 ReasBook method matrices

Table 14 reports the complete LLM and structured-method matrix on the 432-group structured-trace corpus. For methods evaluated with two independent calls, each table entry is the arithmetic mean of the two call-level AUCs; single-call methods retain their single result. Table 15 reports lexical, embedding, and endpoint-only comparisons. Full GCDA obtains five-type MacroAUC/WorstAUC 0.9398/0.8541 with strong separation across the five process changes.

The 1,001 process-preserving ReasBook rows have mean GCDA score 0.6507. The mean score of each negative type is 0.3800 for alternative strategy, 0.5058 for circular dependency, 0.4384 for invalid rewrite, 0.5475 for required bridge removed, and 0.5549 for wrong dependency. These means are descriptive within the ReasBook scale, not calibrated probabilities. AUC remains the primary same-theorem ranking statistic.

H.2 miniF2F complete results

The miniF2F evaluation contains 238 groups and 1,783 rows. The original, faithful-paraphrase, and nonrequired-removal means are 0.6080, 0.4081, and 0.3804, giving an all-positive mean of 0.4730. GCDA obtains pooled, within-theorem, and theorem-equal AUC 0.7689, 0.8782, and 0.8840, re-

Table 14: ReasBook theorem-equal AUC for LLM and structured methods.

Method	Alternative	Cycle	Invalid	Bridge	Wrong dep.	Macro	Worst
GCDA	0.9618	0.9853	0.9691	0.8541	0.9288	0.9398	0.8541
Direct scalar	0.9618	0.5073	0.9009	0.5642	0.4899	0.6848	0.4899
Direct rubric	0.9284	0.5034	0.7792	0.5254	0.5133	0.6499	0.5034
ProofScore	0.9296	0.4888	0.8968	0.5249	0.5195	0.6719	0.4888
LeanScorer-proof	0.9072	0.5085	0.9157	0.5582	0.4967	0.6773	0.4967
Holistic components	0.9055	0.4778	0.8724	0.5231	0.4818	0.6521	0.4778
Holistic formula	0.9008	0.4704	0.8600	0.5246	0.4762	0.6464	0.4704
One-way structured	0.9334	0.5034	0.8025	0.4882	0.5034	0.6462	0.4882

Table 15: ReasBook theorem-equal AUC for lexical, semantic, and endpoint-only methods.

Method	Alternative	Cycle	Invalid	Bridge	Wrong dep.	Macro	Worst
BLEU	0.4579	0.5363	0.6159	0.9321	0.5363	0.6157	0.4579
chrF	0.4583	0.5586	0.6233	0.8862	0.5586	0.6170	0.4583
ROUGE-1	0.6335	0.3772	0.3904	0.8029	0.3772	0.5163	0.3772
ROUGE-2	0.6335	0.4153	0.5343	0.5653	0.4153	0.5127	0.4153
ROUGE-L	0.6624	0.3801	0.4624	0.7526	0.3801	0.5275	0.3801
E5 step F1	0.7130	0.5126	0.4780	0.5348	0.5126	0.5502	0.4780
ROSCOE-style	0.7160	0.5288	0.5228	0.6151	0.5288	0.5823	0.5228
BARTScore	0.7110	0.3498	0.6192	0.6239	0.3498	0.5307	0.3498
BERTScore	0.6238	0.4080	0.4414	0.4281	0.4082	0.4619	0.4080
MoverScore	0.4228	0.5563	0.4479	0.7811	0.5563	0.5529	0.4228
LeanSearch-proof	0.5282	0.4762	0.5664	0.5780	0.4762	0.5250	0.4762
Statement-only controls	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000

spectively. Table 16 gives per-type statistics. The external comparison contains 23 input-matched score series and 41,009 method–variant records under the same fixed theorem-group manifest and input boundary.

H.3 Cross-corpus five-type aggregation

For each process change, we weight corpus AUCs by the number of valid theorem groups and then average the five types. GCDA obtains alternative 0.9197, cycle 0.9794, invalid rewrite 0.9530, bridge removal 0.8428, and wrong dependency 0.8972, for MacroAUC/WorstAUC 0.9184/0.8428. Direct scalar obtains five-type MacroAUC 0.6834. This aggregation preserves the corpus-specific results and uses the same group exclusion throughout.

H.4 Alternative-strategy construction audit

The miniF2F evaluation contains one validated alternative-strategy row for each of its 238 theorem groups. A candidate is admitted only when it remains a mathematically correct proof while changing a principal lemma, construction, branch structure, or governing proof principle. Alternative strategy is therefore positive for mathematical correctness and negative only for fidelity to the specified route. Variant identifiers and content hashes are recorded in the frozen evaluation manifest.

H.5 Complete ablations

Tables 19 and 20 report leave-one-component-out and aggregation-sensitivity results computed from the stored component outputs associated with the reported score artifact. The component sources are synchronized with the current full GCDA score for each row. They are post-hoc recombinations of available component scores, rather than separately rerun end-to-end pipelines, and are therefore interpreted as sensitivity analyses rather than causal estimates of removing an entire pipeline stage.

Table 16: miniF2F GCDA statistics by process change.

Process change	Mean	Rows	Pos.	Neg.	Pooled	Within	Theorem-equal	95% CI
Alternative strategy	0.3202	894	656	238	0.7558	0.8354	0.8431	[0.8039,0.8789]
Circular dependency	0.2496	890	656	234	0.8444	0.9675	0.9687	[0.9544,0.9822]
Invalid rewrite	0.2762	894	656	238	0.8211	0.9207	0.9237	[0.8957,0.9496]
Required bridge removed	0.3426	839	656	183	0.7176	0.8154	0.8197	[0.7814,0.8552]
Wrong dependency	0.3566	890	656	234	0.6937	0.8390	0.8390	[0.8056,0.8725]

Table 17: Complete miniF2F aggregate comparison; the 95% CI for GCDA theorem-equal AUC is [0.8681, 0.8998].

Method	Pooled	Within	Theorem-equal	Macro	Worst
GCDA	0.7689	0.8782	0.8840	0.8788	0.8197
Direct scalar	0.6662	0.6824	0.6885	0.6818	0.5832
Direct rubric	0.6327	0.6411	0.6471	0.6400	0.5776
ProofScore	0.6546	0.6638	0.6675	0.6615	0.5538
LeanScorer-proof	0.6113	0.6159	0.6224	0.6199	0.5534
LeanSearch-proof	0.5049	0.5276	0.5254	0.5334	0.3193
Holistic components	0.6040	0.6449	0.6481	0.6439	0.5687
Holistic formula	0.5941	0.6367	0.6401	0.6358	0.5636
One-way structured	0.6019	0.6178	0.6246	0.6158	0.5413
E5 step F1	0.5137	0.4215	0.4114	0.4102	0.2219
ROSCOE-style	0.5521	0.5294	0.5284	0.5260	0.4013
BARTScore	0.5253	0.5006	0.4934	0.5001	0.3259
BERTScore	0.5233	0.4798	0.4743	0.4762	0.3978
MoverScore	0.5397	0.4807	0.4785	0.4851	0.3424
BLEU	0.5594	0.6030	0.6074	0.6149	0.5274
chrF	0.5556	0.6054	0.6087	0.6172	0.4489
ROUGE-1	0.5617	0.4636	0.4609	0.4762	0.3109
ROUGE-2	0.5655	0.5000	0.4975	0.5076	0.3768
ROUGE-L	0.5641	0.4840	0.4807	0.4958	0.3173

H.6 Negative-type overlap audit

Wrong-dependency variants can also introduce a directed cycle. This overlap occurs in 364 of 431 current ReasBook variants (84.5%) and in 24 of 234 miniF2F variants (10.3%), leaving 67 and 210 acyclic variants, respectively. The complete current-population AUCs are 0.9288 and 0.8390. They reflect both direction and cycle sensitivity. This analysis uses the frozen evaluation manifest without selecting groups by cycle status or observed AUC.

Bridge removal has a different confound. Deleting an intermediate step changes proof length and token overlap, allowing lexical metrics to detect a surface-level deletion without determining whether the removed step is logically necessary. BLEU consequently reaches 0.9321 on ReasBook and ROUGE-L reaches 0.7746 on miniF2F. Their substantially weaker performance on wrong dependency, where the dependency edge changes while the proof text remains largely unchanged, supports this interpretation.

H.7 Stochastic stability

Independent calls of the same external method show modest but nonzero variation. On the input-matched miniF2F population, theorem-equal AUC ranges from 0.6744 to 0.7026 for Direct scalar and from 0.6152 to 0.6341 for One-way structured. On ReasBook, the corresponding ranges are narrower for the evaluated baselines. These ranges measure stochastic call variation; repeated calls are not treated as separate methods.

We assess stochastic variability using paired outputs for 165 ReasBook theorem groups and 1,214 matched scoring rows in the current evaluation population. Every language-model-dependent stage was rerun with the same model, prompts, validators, formal evidence, obligations, thresholds, and scoring formula, without reusing earlier responses. Both runs are compared on the same theorem

Table 18: Complete miniF2F theorem-equal AUC matrix.

Method	Alternative	Cycle	Invalid	Bridge	Wrong dep.	Macro	Worst
GCDA	0.8431	0.9687	0.9237	0.8197	0.8390	0.8788	0.8197
Direct scalar	0.8181	0.5832	0.7922	0.6068	0.6088	0.6818	0.5832
Direct rubric	0.7619	0.5778	0.6984	0.5776	0.5840	0.6400	0.5776
ProofScore	0.8015	0.5538	0.7850	0.5915	0.5759	0.6615	0.5538
LeanScorer-proof	0.6054	0.5559	0.7675	0.6175	0.5534	0.6199	0.5534
LeanSearch-proof	0.3193	0.5313	0.6723	0.6129	0.5313	0.5334	0.3193
Holistic components	0.6658	0.5687	0.7908	0.6122	0.5817	0.6439	0.5687
Holistic formula	0.6572	0.5636	0.7812	0.6006	0.5766	0.6358	0.5636
One-way structured	0.7337	0.5413	0.7094	0.5492	0.5452	0.6158	0.5413
E5 step F1	0.9104	0.2219	0.2990	0.3980	0.2219	0.4102	0.2219
ROSCOE-style	0.8704	0.4241	0.4013	0.5100	0.4241	0.5260	0.4013
BARTScore	0.7675	0.3259	0.4818	0.5993	0.3259	0.5001	0.3259
BERTScore	0.6618	0.3978	0.4328	0.4909	0.3978	0.4762	0.3978
MoverScore	0.7598	0.3551	0.3424	0.6134	0.3551	0.4851	0.3424
BLEU	0.6127	0.5274	0.6520	0.7550	0.5274	0.6149	0.5274
chrF	0.4489	0.6207	0.6345	0.7614	0.6207	0.6172	0.4489
ROUGE-1	0.6674	0.3109	0.3487	0.7432	0.3109	0.4762	0.3109
ROUGE-2	0.6922	0.3768	0.4226	0.6694	0.3768	0.5076	0.3768
ROUGE-L	0.6982	0.3173	0.3718	0.7746	0.3173	0.4958	0.3173

Table 19: ReasBook five-type component and aggregation ablations.

ReasBook setting	Alternative	Cycle	Invalid	Bridge	Wrong dep.	Macro	Worst
Full GCDA	0.9618	0.9853	0.9691	0.8541	0.9288	0.9398	0.8541
Without M_N	0.9556	0.9853	0.9699	0.8770	0.9261	0.9428	0.8770
Without M_F	0.9514	0.9865	0.9676	0.8254	0.9273	0.9316	0.8254
Without Q	0.9456	0.9780	0.7390	0.6442	0.9134	0.8440	0.6442
Without R	0.9595	0.9780	0.9676	0.8642	0.9227	0.9384	0.8642
Without C	0.9618	0.7053	0.9691	0.8541	0.6667	0.8314	0.6667
M_N only	0.9236	0.4938	0.7458	0.5040	0.5112	0.6357	0.4938
M_F only	0.8256	0.6212	0.5100	0.7762	0.6239	0.6714	0.5100
Geometric aggregation	0.9201	0.9302	0.9211	0.8159	0.8807	0.8936	0.8159
Harmonic aggregation	0.9178	0.9283	0.9211	0.7925	0.8765	0.8872	0.7925
Minimum aggregation	0.9120	0.9012	0.9174	0.7099	0.8140	0.8509	0.7099
Product aggregation	0.9201	0.8575	0.9211	0.8159	0.7863	0.8602	0.7863

groups and scoring rows; context-only rows are omitted from the metrics. These agreement statistics are conditional on paired, schema-valid outputs and do not estimate end-to-end execution reliability.

Table 21: Paired GCDA repeat analysis on 165 ReasBook groups in the current evaluation population.

Statistic	Value
Score Pearson / Spearman	0.8188 / 0.8222
Spearman 95% CI	[0.7792, 0.8603]
Theorem-demeaned Pearson / Spearman	0.8885 / 0.8836
Mean / P95 absolute difference	0.0617 / 0.2033
Within-theorem ordering agreement	0.9177
MacroAUC across runs	0.9404 / 0.9263
WorstAUC across runs	0.8596 / 0.8758
Mean per-proof core Jaccard	0.7700
Uncovered-required-region Jaccard	0.6817

The theorem-demeaned correlations and 91.77% ordering agreement indicate greater stability for within-theorem comparisons than for corpus-wide score calibration. The P95 absolute difference is 0.2033. The deterministic structural-soundness term is identical across runs; learned-component Spearman correlations range from 0.6813 to 0.7682. The score Spearman interval uses 2,000

Table 20: miniF2F five-type component and aggregation ablations.

miniF2F setting	Theorem-equal	Alternative	Cycle	Invalid	Bridge	Wrong	Macro	Worst
Full GCDA	0.8840	0.8431	0.9687	0.9237	0.8197	0.8390	0.8788	0.8197
Without M_N	0.8851	0.7875	0.9701	0.9324	0.8575	0.8650	0.8825	0.7875
Without M_F	0.8807	0.8088	0.9694	0.9321	0.8315	0.8412	0.8766	0.8088
Without Q	0.7974	0.8158	0.9665	0.7416	0.7195	0.7208	0.7928	0.7195
Without R	0.8763	0.8354	0.9480	0.9237	0.8169	0.8312	0.8710	0.8169
Without C	0.8543	0.8431	0.8405	0.9237	0.8197	0.8262	0.8506	0.8197
M_N only	0.6921	0.7689	0.6724	0.7255	0.6466	0.6211	0.6869	0.6211
M_F only	0.6754	0.7006	0.6428	0.6744	0.7140	0.6595	0.6783	0.6428
Geometric aggregation	0.7883	0.7588	0.8223	0.8368	0.7587	0.7525	0.7858	0.7525
Harmonic aggregation	0.7794	0.7496	0.8152	0.8340	0.7404	0.7411	0.7761	0.7404
Minimum aggregation	0.7530	0.7377	0.7995	0.8228	0.6958	0.6863	0.7484	0.6863
Product aggregation	0.7846	0.7588	0.8038	0.8368	0.7587	0.7525	0.7821	0.7525

theorem-group bootstrap replicates (seed 20260925). Core Jaccard first averages over natural transitions within each proof and then equally over proofs; uncovered-region Jaccard is also averaged over proofs.

The archived GPT-5.6-assisted graph-quality audit defined a 100-group high-evidence subset. Its intersection with the current evaluation population contains 84 groups and 624 rows. On these groups, Pearson/Spearman correlations are 0.8141/0.8091, mean absolute difference is 0.0618, and within-theorem ordering agreement is 0.9216. This is an AI-assisted sensitivity subset, not an independent expert-label validation.

For the input-matched comparison, every external score series covers the same frozen 1,783-row manifest. The final assembly contains $23 \times 1{,}783 = 41{,}009$ records, all of which pass schema and coverage validation. Every method is evaluated on the exact proof text and formal evidence identified by that manifest under its declared input boundary. Formula ablations use stored GCDA component outputs and incur no additional model calls.

H.8 Diagnostic localization

We evaluate whether GCDA's stored diagnostics identify the edited structure, separately from proof-level ranking. This post-hoc analysis uses the previously frozen diagnostic annotations for the 432 current ReasBook groups and makes no new model calls. Diagnostics retain their original run provenance; they are not newly generated outputs for later score overlays. Gold targets are available only when the construction record identifies a unique informal transition, dependency, or frozen required region; unavailable cases are not imputed.

Table 22: ReasBook diagnostic localization. Top-1 is exact dependency accuracy for wrong dependency, exact-set accuracy for cycles, and Recall@1 for transition or required-region targets.

Target	Avail.	Top-1	Recall@3	MRR
Required bridge / region	143/378	0.6364	0.9371	0.7969
Wrong dependency (acyclic)	67/431	0.0448	0.1194	0.1910
Circular dependency	431/431	1.0000	1.0000	1.0000
Invalid rewrite transition	432/432	0.8958	0.9375	0.9246

Bridge targets are defined only when the removed informal transition was associated with a frozen required region. Wrong-dependency localization excludes 364 edits that also create a cycle, and its Top-1 tie rate is 0.8657; the low exact score should therefore be read together with Recall@3 and MRR. Circular-dependency localization deterministically replays the same strongly connected component rule used by the structural-soundness term, so it measures consistency of the supplied structural certificate rather than learned recovery of a hidden dependency from prose. Invalid-rewrite localization uses contradiction identifiers and transition-level claim–Delta diagnostics. These results emphasize that ranking a corrupted proof and localizing the edited structure are distinct capabilities; in particular, exact localization of acyclic wrong-dependency edits remains challenging.

H.9 EXPERT ASSESSMENT OF PROOF-ROUTE CORRESPONDENCE

Materials and annotation protocol. The annotation packet contains 50 distinct natural-language–Lean proof pairs: 48 from analysis texts and two from optimization material. This is a convenience sample, rather than a random sample across mathematical domains. Each item provides the natural statement and proof, the complete target Lean declaration, and supporting definitions and lemma proofs. The two packet orders share anonymous identifiers and mathematical content; automatic scores, anticipated labels, and other reviewers' judgments are absent. Supporting declarations are reference material, and only those actually used contribute to the proof route. The packet provides bounded supporting evidence, not a complete dependency closure or a new compilation audit of all 50 excerpts.

The instructions ask reviewers to determine whether the proofs reach their conclusion through the same main mathematical argument, allowing notation changes and verifiable routine compression. Reviewers first check statement compatibility, then compare core methods, key intermediate conclusions, and their dependencies. Scores mean: 3, the same core route with only routine differences; 2, the same core route with local differences; 1, recognizable common steps but substantive differences in a key argument or dependency; and 0, substantially different core strategies. Short `simp` or `ring` proofs are not penalized for brevity, while a decisive theorem call requires checking its supplied proof. Each annotation includes both route summaries, line-level evidence, a rationale, and confidence. The final annotation requirement assigns a numeric 0–3 rating to every pair and records statement compatibility separately. The initial assessment instructions request independent judgments without model-generated ratings or discussion before submission.

Review stage and statistical definition. The available numerical comparison uses review A (`RV-Q7-R2`) and review B (`R-Q7M2`), aligned by pair identifier. Review A records revision after access to the other reviewer's judgments. Accordingly, the statistics below describe *agreement between numerical-review versions*, not agreement between the original independent blind submissions. Both versions contain numeric ratings for all 50 pairs under the final annotation requirement. The primary comparison includes all 50 pairs; a supplementary analysis restricts attention to the subset passing both reviewers' statement-compatibility checks. No score is changed or adjudicated in this analysis.

For ratings $a, b \in \{0, 1, 2, 3\}$, linear-weighted Cohen's κ uses agreement weights $w_{ab} = 1 - |a-b|/3$ and the observed reviewer marginals to calculate chance agreement. Exact agreement and linear-weighted κ are the primary ordinal summaries. The binary grouping combines 2–3 (at least broadly corresponding) and 0–1 (at most partially corresponding); it summarizes the recorded ratings without creating adjudicated labels.

Table 23: Agreement between numerical-review versions. Access to the other reviewer's judgments precedes revision of review A. The 46-pair subset excludes any pair marked nonmatching or uncertain by either reviewer.

Statistic	All numerical ratings	Both statements compatible
Proof pairs	50	46
Exact ordinal agreement	$43/50$ (86.0%)	$39/46$ (84.8%)
Linear-weighted κ	0.834	0.826
Binary agreement	$46/50$ (92.0%)	$42/46$ (91.3%)
Binary κ	0.752	0.747

Table 24: Ordinal rating contingency table for all 50 pairs; rows are review A and columns are review B.

A \ B	0	1	2	3	Total
0	2	0	0	0	2
1	0	6	0	0	6
2	0	4	20	2	26
3	0	0	1	15	16
Total	2	10	21	17	50

All seven disagreements are one level apart. Supplementary full-sample statistics are unweighted $\kappa = 0.784$, quadratic-weighted $\kappa = 0.891$, and mean absolute difference 0.14. Pearson and Spearman correlations are 0.897 and 0.883; these measure association and do not replace agreement statistics. Under the binary grouping, both reviews rate 38 pairs at least broadly corresponding and eight at most partially corresponding; the remaining four receive 2 from A and 1 from B.

Disagreements and statement scope. Four disagreements cross the 1/2 boundary: EP012 (metric versus induced-topology continuity), EP021 and EP047 (explicit counterexample sequences versus closure tools), and EP027 (squared-modulus reasoning versus a triangle-inequality argument). The three 2/3 disagreements are EP001, EP014, and EP018, concerning norm estimates, reciprocal-limit reasoning, and continuity under codomain restriction. The recorded rationales differ in how strongly these changes affect the main route. These descriptions summarize the reviews; they are not new adjudications.

Statement-compatibility judgments agree on $47/50$ pairs (94%). Both reviews mark EP003 as `different`. For EP020 and EP045, A marks `uncertain` and B marks `different`; for EP032, A marks `different` and B marks `compatible`. Their route ratings are 2/2, 2/2, 3/3, and 2/2, respectively. These scope flags are separate from the numeric route ratings. As a supplementary sensitivity analysis, restricting attention to the 46 pairs both reviewers marked compatible gives $39/46$ exact agreement (84.8%) and linear-weighted $\kappa = 0.826$, with binary agreement $42/46$ (91.3%). All 50 pairs remain in the primary analysis.

Interpretation. This small study describes how closely the reviewed expert judgments agree and identifies the remaining rubric boundaries. The sample concentration and review-stage dependence limit generalization. No GCDA–expert correlation, ranking accuracy against expert labels, or superiority over a direct judge is established by inter-review agreement alone. Such claims require a separate input-matched comparison with preserved expert labels and a documented treatment of statement-scope differences.

H.10 Interpretation boundary

Controlled variants isolate specific process properties but do not reproduce the full error distribution of deployed autoformalization systems. Natural state-transition structures may admit multiple reasonable decompositions, and opaque formal automation limits observability. The score is best used for same-theorem screening, ranking, and diagnosis. It does not replace kernel verification, and a low score does not imply that the formal theorem is false.

I Case Studies and Local Diagnostics

We present six paired cases from the frozen ReasBook evaluation to show how the GCDA components respond to qualitatively different failures. These cases are explanatory rather than an additional statistical evaluation: they were chosen to isolate local semantic alignment, omitted bridges, reverse formal coverage, dependency direction, cyclic support, and reference-conditioned route fidelity. Each pair uses one verified Lean proof, an aligned natural proof, and one controlled negative variant. All scores and local diagnostics below are read from the frozen run; no case was rescored for this analysis.

Table 25 gives the complete component vectors. Scores are rounded to three decimals and satisfy $S = C(M_N + M_F + Q + R)/4$ before rounding. Because S is an alignment score rather than a calibrated probability, the relevant evidence is the paired response within a theorem and the component that explains it. In the Lean excerpts, TeX escapes only typeset mathematical glyphs; the proof steps and theorem names are those used by the verified source proof.

Evidence audit. The displayed proof accounts are faithful, layout-conscious condensations of the frozen informal-proof JSONL; the full verbatim texts remain in the artifact. The component values, obligation credits, and localization ranks come from the corresponding frozen score and diagnostic logs. Formal proofs are resolved through the provenance manifest to a commit-pinned source snapshot. For all six groups, that manifest records a kernel-verified Lean proof, no `sorry` or `admit`, successful LSP trace extraction, and successful scoring; the recorded source hashes also match the

pinned files. None of the groups intersects the later adaptive score overlay. Thus the case analysis combines already-produced artifacts rather than new judgments made after seeing the displayed score differences.

Manual validity checks. For every positive, we checked that the main claims, required hypotheses, and route-defining Lean steps agree in the stated order. For every negative, we checked the exact reason it is non-aligned. Two negatives (`::320` and `::0`) deliberately preserve the surface proof and corrupt only its explicit dependency graph, so their defect must be assessed structurally. The Vitali argument in `::36` is mathematically correct but negative for reference-route fidelity. Finally, the bridge omission in `::127` is detected at the aggregate and obligation levels even though its proof-level localization is diffuse; we report that limitation explicitly.

I.1 A SEMANTIC CONTRADICTION WITH PRESERVED FORMAL COVERAGE

Aligned natural proof (`::173`, `ae_eq_iff`). For filters, equality is equivalent to mutual order: $F = G$ exactly when $F \le G$ and $G \le F$. Applying this to the almost-everywhere filters gives $\mathrm{ae}\,\mu = \mathrm{ae}\,\nu$ iff both corresponding order relations hold. The standard characterization rewrites these relations as $\mu \ll \nu$ and $\nu \ll \mu$, respectively. Substitution therefore yields $\mathrm{ae}\,\mu = \mathrm{ae}\,\nu \leftrightarrow (\mu \ll \nu \wedge \nu \ll \mu)$.

Controlled negative proof. The negative repeats the same argument but concludes $\mathrm{ae}\,\mu \neq \mathrm{ae}\,\nu \leftrightarrow (\mu \ll \nu \wedge \nu \ll \mu)$. Thus its last sentence contradicts both the theorem and its own opening equality characterization; this is a semantic error rather than a paraphrase.

Formal proof. The verified Lean proof implements exactly the equality characterization and the two directional absolute-continuity rewrites:

```
theorem ae_eq_iff {μ ν : Measure Omega} :
    ae μ = ae ν ↔ μ ≪ ν ∧ ν ≪ μ := by
  rw [le_antisymm_iff, ae_le_iff_absolutelyContinuous,
      ae_le_iff_absolutelyContinuous]
```

Why the score is informative. The positive proof is aligned sentence by sentence with the three rewrites. For the negative, M_N and Q both fall to zero, while M_F remains high ($.940 \to .946$) because the formal region is still mentioned. The score therefore identifies a wrong natural conclusion without misdiagnosing a missing Lean region. The edited conclusion node `N4` is localized at rank one (Recall@1 and MRR both 1).

I.2 AN OMITTED ALGEBRAIC BRIDGE WITH THE FINAL CLAIM RETAINED

Aligned natural proof (`::127`, `mu_sq_identity`). Let $T = \sigma_1\sigma_2 D_2$. Positivity of σ_1, σ_2 and the hypotheses on D_2 imply $T > 0$, so $D_1/T \ge 0$ and $(\sqrt{D_1/T})^2 = D_1/T$. Substituting the definition of μ gives $\mu^2 = (2A/(N+1))^2(D_1/T)$. Since T and $N+1$ are nonzero, multiply by $T(N+1)^2$, cancel the denominators, and normalize to obtain $\mu^2 T(N+1)^2 = 4A^2 D_1$. Substituting the definition of T proves the target.

Controlled negative proof. The negative keeps the square-root calculation and copies the final target, but deletes the sentence that multiplies by $T(N+1)^2$, invokes the nonzero hypotheses, and cancels the denominators. The displayed formula for μ^2 does not imply the copied conclusion without this transformation, so the negative contains a genuine unsupported jump.

Formal proof. After proving the square identity, the Lean proof makes the omitted bridge explicit through nonzero witnesses, `field_simp`, and `nlinarith`:

```
have hσ1ne : σ1 ≠ 0 := ne_of_gt hσ1
have hσ2ne : σ2 ≠ 0 := ne_of_gt hσ2
have hN1ne : ((N : ℝ) + 1) ≠ 0 := by nlinarith
field_simp [hσ1ne, hσ2ne, hD2ne, hN1ne] at hμ_sq
nlinarith [hμ_sq]
```

Why the score is informative. Removing the natural bridge lowers Q from .920 to .190 and the total score from .606 to .345, despite preserving the final sentence. The associated formal obligation `O006` loses all credit ($.902 \rightarrow 0$), showing that endpoint overlap cannot substitute for a derivation. The aggregate detection is stronger than its localization: `O006` is only rank nine (MRR .111; uncovered-set recall 1.0 and precision .091), which is a limitation of this example rather than evidence of exact localization.

I.3 Loss of the exact Lean-side bridge

Aligned natural proof (`::457`, unit reflection). A ring homomorphism sends units to units. Conversely, surjectivity of the map on prime spectra upgrades the homomorphism to an `IsLocalHom`; for a local homomorphism, `isUnit_map_iff` reflects units. Reversing that equivalence gives $\mathrm{IsUnit}(x) \leftrightarrow \mathrm{IsUnit}(\varphi(x))$.

Controlled negative proof. The negative retains a mathematically plausible prime-ideal explanation and the sentence constructing `IsLocalHom`, but deletes the invocation of `isUnit_map_iff`. Its Lean-specific account consequently jumps from the instance to the reversed equivalence without explaining the formal step that actually proves it.

Formal proof. The complete verified proof of `isUnit_iff_isUnit_map_of_comap_surjective` has only the two route-critical steps:

```
1 haveI : IsLocalHom φ :=
2   IsLocalHom.of_comap_surjective φ hφ
3 exact (isUnit_map_iff φ x).symm
```

Why the score is informative. The positive tail is directly aligned with both formal steps. In the negative, the two obligation credits fall from .9745 and .9820 to zero, and M_F falls from .978 to zero, whereas Q remains .820. Thus GCDA does not declare the prose meaningless; it identifies that the prose no longer covers the specified proof route. The removed bridge `N10` and its obligation `O001` are localized at rank one.

I.4 Correct sentences with a reversed dependency

Aligned natural proof (`::320`, local ODE existence). First invoke the stronger local existence-and-uniqueness theorem for a higher-order holomorphic ODE. Under $k > 0$ and analyticity of F, this theorem produces a function φ, a proof h_φ that its germ is a local solution, and a uniqueness statement (step `N2`). The target asks only for existence, so return φ and h_φ and discard uniqueness (step `N3`). The aligned dependency is `N2`→`N3`.

Controlled negative proof. The negative contains the same four natural sentences but records the edge `N3`→`N2`. This is not a stylistic reordering: `N3` consumes the witness and proof produced by `N2`, so it cannot be their premise.

Formal proof. The Lean data flow is explicit. It applies `exists_eventuallyEq_unique_solution_of_higher_order_holomorphic_ode`; the following normalized excerpt makes the produced triple compact enough for one column. The `rcases` step produces the two required objects before `exact` returns them.

```
1 rcases h_exists_unique hk hF with
2   ⟨φ, h_φ, _⟩
3 exact ⟨φ, h_φ⟩
```

Why the score is informative. The sentence-level and coverage terms remain nearly unchanged, while R falls from .618 to .382. For the controlled edge, formal forward/backward reachability is $(1, 0)$ in the positive direction and $(0, 1)$ after reversal, so the edge-direction score changes from 1 to 0. The corrupted edge is localized exactly at rank one. The smaller aggregate change ($.707 \rightarrow .653$) is therefore interpretable rather than a failure to notice the structural edit.

I.5 Circular support hidden by otherwise competitive components

Aligned natural proof (`::0`, Lebesgue axioms). Take $\mathcal{M}$ to be the Lebesgue-measurable subsets of $\mathbb{R}^n$ and m to be volume. Open-set measurability, closure under complement and countable unions and intersections, zero measure of the empty set, ENNReal bounds, countable additivity, and the unit-cube formula establish the first axioms in order. Translation measurability and invariance establish the last axiom (step `N7`); these facts together yield the final conjunction (step `N8`). In particular, `N7`→`N8`.

Controlled negative proof. The negative preserves every sentence and positive edge, but adds `N8`→`N7`. The final conjunction now supports one of the premises already used to prove that conjunction, creating the two-node cycle `N7`↔`N8`. This graph-level artifact is therefore circular even though its surface proof is unchanged.

Formal proof. The verified proof constructs measurable sets and volume, discharges the axioms with nested constructors, and ends by proving translation invariance; the final formal region is:

```
intro Omega x hOmega
refine ⟨?_,?_⟩
· simpa [Set.image_add_left] using
    (show MeasurableSet Omega from hOmega).preimage
      (measurable_const_add (-x))
· simp [Set.image_add_left]
```

Why the score is informative. SCC replay recovers exactly the inserted cycle (precision, recall, and F1 all 1). Without the structural factor, the negative component mean is .674, and R even increases. The cycle factor $C = .857$ reduces the final score to .578, below the aligned score .667. This case isolates the reason for a multiplicative structural-soundness term: ordinary semantic or edge-agreement components could otherwise compensate for explicit circular support.

I.6 A correct proof following a different route

Aligned natural proof (`::36`, a nonmeasurable set). Assume for contradiction that every subset of $\mathbb{R}$ is measurable. A countable generating family has cardinality at most the continuum, so the generated measurable sets also have cardinality at most the continuum. Under the assumption, the full power set injects into the subtype of measurable sets, hence $|\mathcal{P}(\mathbb{R})| \le \mathfrak{c}$. But $|\mathcal{P}(\mathbb{R})| = 2^{\mathfrak{c}}$ and Cantor's theorem gives $\mathfrak{c} < 2^{\mathfrak{c}}$, a contradiction. Therefore a nonmeasurable subset exists.

Controlled negative proof. The alternative instead chooses a Vitali representative set $V \subseteq [0,1]$ modulo rational differences. The translates $V + q$ for $q \in \mathbb{Q} \cap [-1,1]$ are pairwise disjoint, and their union U satisfies $[0,1] \subseteq U \subseteq [-1,2]$. If V had measure zero, countable additivity would make the union null; if V had positive measure, the union would have infinite measure. Both conclusions contradict the two containments, so V is nonmeasurable. This is a valid and complete mathematical proof, but it realizes none of the cardinality route's formal obligations.

Formal proof. The verified Lean proof follows the cardinality route, culminating in the power-set identity and Cantor contradiction:

```
have hSetCardEq : Cardinal.mk (Set ℝ) =
    2 ^ Cardinal.continuum := by
  calc
    Cardinal.mk (Set ℝ) = 2 ^ Cardinal.mk ℝ :=
      Cardinal.mk_set (alpha := ℝ)
    _ = 2 ^ Cardinal.continuum := by rw [Cardinal.mk_real]
have hContinuumLtSet :
    Cardinal.continuum < Cardinal.mk (Set ℝ) := by
  rw [hSetCardEq]
  exact Cardinal.cantor Cardinal.continuum
exact (not_le_of_gt hContinuumLtSet) hSetCardLeContinuum
```

Why the score is informative. The positive proof follows the Lean obligations in order. For the Vitali proof, all ten natural-node matches and all five required formal-region credits are zero; M_N

and M_F therefore fall to zero and S falls from .777 to .145. The frozen labels separately record that this variant is mathematically correct, complete, and dependency coherent, but strategy different and trace unfaithful. The case thus demonstrates the intended scope of GCDA: it evaluates correspondence to a specified formal process, not mathematical correctness in isolation.

Table 25: GCDA component scores for the six case-study pairs. "Negative" means non-aligned with the specified formal route; the alternative-strategy variant in the last pair is mathematically correct.

Case	Variant	M_N	M_F	Q	R	C	S
`::173`	Positive	.659	.940	.950	.862	1.000	.853
	Negative	.000	.946	.000	.500	1.000	.361
`::127`	Positive	.560	.324	.920	.621	1.000	.606
	Negative	.366	.266	.190	.557	1.000	.345
`::457`	Positive	.367	.978	.930	.500	1.000	.694
	Negative	.319	.000	.820	.500	1.000	.410
`::320`	Positive	.524	.805	.880	.618	1.000	.707
	Negative	.550	.812	.870	.382	1.000	.653
`::0`	Positive	.639	.628	.890	.510	1.000	.667
	Negative	.613	.620	.800	.661	.857	.578
`::36`	Positive	.652	.894	.860	.703	1.000	.777
	Negative	.000	.000	.080	.500	1.000	.145

I.7 What the cases establish

The cases provide three complementary checks on score behavior. First, a content error can lower M_N and Q while leaving formal coverage high (`::173`); conversely, missing route coverage can lower M_F without forcing a low coherence judgment (`::457`). Second, explicit omission, direction, and cycle diagnostics expose failures that endpoint or sentence overlap can miss (`::127`, `::320`, and `::0`). Third, the route-different case separates proof correctness from reference-conditioned alignment (`::36`). Together, these paired decompositions explain why the aggregate should be read with its component vector and local certificate, not as an unsupported scalar judgment.

I.8 A worked scoring chain: preserving periodicity under coercion

This additional walkthrough uses the original proof for ReasBook group 24 from one archived execution (`sample200`, run 1, proof ID `P7`). All intermediate values and the final score below come from that execution, rather than from the six-pair summary above. It illustrates the recorded scoring procedure, not an additional benchmark result.

Proof pair and natural records. Let $g : \mathbb{R} \to \mathbb{R}$ have period T. The natural proof assumes this periodicity (N0), fixes arbitrary x (N1), derives $g(x + T) = g(x)$ (N2), applies the inclusion $\mathbb{R} \to \mathbb{C}$ to obtain equality of the coerced values (N3), identifies the required pointwise identity (N4), and concludes periodicity because x was arbitrary (N5). The reference is:

```
lemma
  helperForExample_5_1_2_realToComplex_periodic
    {g : ℝ → ℝ} {T : ℝ}
    (hg : Function.Periodic g T) :
    Function.Periodic
      (fun x : ℝ => (g x : ℂ)) T := by
  intro x
  exact congrArg
    (fun y : ℝ => (y : ℂ)) (hg x)
```

The recorded natural graph includes claim-use and scope links from N0/N1 to N2, from N2 to N3, and through N4 to N5. Auxiliary entity-reference and textual-order edges are not all independent mathematical dependencies. In Stage A, N3 is classified as a rewrite with importance 0.84: its input is the real equality, its expected effect is preservation of equality by coercion, and its output is the complex equality required by the target. The archived Stage A request already contains the natural graph; it is not the original raw-text extraction request.

Formal Deltas and frozen obligation. There are two explicit tactic actions. The `intro x` card introduces $x : \mathbb{R}$ and exposes the pointwise equality goal. The `exact congrArg` card uses `hg x`, transports equality to $\mathbb{C}$, and closes that goal. Their recorded formal dependency follows goal production and consumption. The trace also contains macro containers, including the whole-proof closure; these must not be counted as additional independently written tactics.

The frozen set contains one obligation, O000: establish periodicity of the coerced function. Its required inputs are g, T, and the periodicity hypothesis; its output is $\mathrm{Periodic}(x \mapsto (g(x) : \mathbb{C}), T)$. The `exact congrArg` action is its core and `intro x` is its internal connector. Thus the obligation describes one mathematical responsibility spanning two tactic actions.

Forward alignment and whole-proof validation. The recorded local importance/score pairs for N0–N5 are $(.64, .16)$, $(.66, .70)$, $(.78, .31)$, $(.84, .70)$, $(.34, .18)$, and $(.97, .70)$. N1 matches `intro x`, N3 matches `exact congrArg`, and N5 cites the whole-proof closure. N0, N2, and N4 receive no independent core in the final validated partition. In particular, deriving the real equality is compressed inside `hg x`, while recognizing the target is not separately written as a tactic. This record shows why a short formal proof need not yield a perfect one-to-one local alignment with six natural sentences.

Stage D returns a judgment for P7: $Q = .9$, a supported conclusion, and no missing critical bridge. Its explanation identifies the route "introduce x, use `hg x`, coerce equality, conclude periodicity." The displayed local assignments are the final validated ones, after conflict handling, rather than unaccepted correction proposals.

Reverse coverage and aggregation. O000 is matched to N5 with input/effect/output/method scores $(.83, .93, .98, .89)$. Applying the fixed weights gives

$$c_0 = .15(.83) + .40(.93) + .35(.98) + .10(.89) = .9285.$$

Both effect and output pass the coverage gate. Since this execution has one frozen obligation, $M_F = .9285$. The importance-weighted forward score is $M_N = 2.1344/4.23 \approx .504586$; the remaining components are $R = .5$ and $C = 1$. Consequently,

$$S_{\mathrm{GCDA}} = \frac{.504586 + .9285 + .9 + .5}{4} \approx .708272.$$

The neutral R records insufficient observable directional evidence under the scoring rule, not verified perfect direction agreement. The difference between forward support and reverse coverage remains visible in the certificate instead of being hidden by the total score.

Artifact correspondence. The archive for group 24 contains the natural packet in the Stage A request, its validated analyses, Stage B formal Delta cards, the final Stage C/D partition, frozen obligations, validated reverse matches, and the result record. These provide a reproducible path through this scoring execution; the walkthrough does not claim to recover missing original text-segmentation calls.