Ground False: Dissecting Errors in Formal Mathematics Benchmarks

One An [*1] **Marcus J. Min** [*1] **Xujie Si** [2] **Osbert Bastani** [1]

Abstract

The ground truths in formal mathematics benchmarks are taken on faith. For theorem proving, released formalizations come without formal proofs that certify their correctness. For autoformalization, faithfulness to the informal source is judged only by the benchmark authors, yet such judgments are inherently subjective and lack collective consensus. We show these assumptions fail at scale: our audit of 367 ProofNet formalizations shows that 104 (28%) are mathematically false, and a further 100 (27%) are not false yet still unfaithful, for a total of 204 (56%) unfaithful items. We dissect these errors along three axes (provability, logical strength, root cause) and show they bias evaluation in predictable, asymmetric directions: false ground truths cap theorem-proving signal, while equivalence-based autoformalization metrics compress model gaps through capping and coincidental agreement. We release the corrected benchmark **ProofNet-Verified**, produced by a semi-automated pipeline combining Lean provability checks, an LLM faithfulness judge, and human review. Applying the same pipeline to six other benchmarks reveals that unfaithful-GT rates span more than an order of magnitude (4.8% to 60.0%), confirming that the failure modes generalize while absolute quality is dominated by curatorial process. Our data and code are available at https://github.com/marcusm117/ProofNet-Verified.

[*]Equal contribution [1]University of Pennsylvania [2]University of Toronto. Correspondence to: One An <onean@seas.upenn.edu>, Marcus Min <marcmin@seas.upenn.edu>, Xujie Si <six@cs.toronto.edu>, Osbert Bastani <obastani@seas.upenn.edu>.

The 3rd *AI for Math Workshop* at the 43rd International Conference on Machine Learning (ICML), Seoul, South Korea, 2026. Copyright 2026 by the author(s).

1. Introduction

The ground truths in formal mathematics benchmarks are taken on faith. Benchmarks such as ProofNet (Azerbayev et al., 2023), PutnamBench (Tsoukalas et al., 2024), and the FATE family (Jiang et al., 2026) drive model comparisons on two interlocking tasks: theorem proving in interactive proof assistants like Lean (Moura & Ullrich, 2021), and *autoformalization*, the translation of mathematical claims from natural language into a formal language. Yet the released ground-truth (GT) formalizations that underpin these evaluations have, to our knowledge, never been collectively verified. For theorem proving, formalizations come without formal proofs that certify their correctness. For autoformalization, faithfulness to the informal source is judged only by the benchmark authors, yet such judgments are inherently subjective and lack collective consensus.

We show these assumptions fail at scale. Our audit of all 367 ProofNet items finds that 204 (56%) carry an unfaithful ground truth, and over half of those are mathematically false. These errors bias evaluation in predictable directions: false GTs cap theorem-proving signal, while equivalence-based autoformalization metrics penalize faithful models more than unfaithful ones. We dissect ground-truth errors along three axes (provability, logical strength, root cause), fix the identified "Ground Falses," and release **ProofNet-Verified**, a corrected benchmark with per-item taxonomy labels for all 367 items. Applying the same pipeline to six other benchmarks (PutnamBench, ProverBench (Ren et al., 2025), CombiBench (Liu et al., 2025), LeanCat (Xu et al., 2025), and the FATE family) reveals unfaithful-GT rates spanning 4.8% to 60.0%, confirming that the failure modes generalize.

Our contributions are: (1) a three-axis **taxonomy** of ground-truth errors in formal-math benchmarks, grounded in a manual audit of ProofNet (Section 2); (2) an **analysis** showing that wrong ground truths bias theorem-proving and autoformalization evaluation in predictable directions, including a gap-compression effect that penalizes faithful autoformalizers (Section 3); (3) a **semi-automated audit pipeline** combining Lean provability checks, an LLM faithfulness judge, and human review (Section 4); (4) **ProofNet-Verified**, a corrected ProofNet with per-item taxonomy labels for all 367

items; and (5) a **cross-benchmark audit** of PutnamBench, ProverBench, CombiBench, LeanCat, and the FATE family, showing the failure modes generalize while error rates vary by an order of magnitude (Section 5).

2. Taxonomy of Errors

Our audit of the 367 ProofNet (Azerbayev et al., 2023) items finds that 204 (56%) carry a ground truth (GT) that does not faithfully encode the natural-language (NL) claim it is paired with. We take a GT to be *faithful* when it is logically equivalent to a ground truth formalization I of the NL claim *modulo an acceptable background theory* T of standard definitions, library lemmas, and conventions; formally, $T \vdash (G \leftrightarrow I)$ for the released GT G. Equivalence modulo T is sufficient for faithfulness; a GT not equivalent to I is unfaithful, and the logical-strength axis below records how (stronger, weaker, or incomparable), while an NL source that is itself ambiguous or mathematically incorrect pins down no I and is a defect of the benchmark rather than of the formalization. To make these errors useful as a tool for assessing what existing benchmarks can and cannot evaluate, we organize them along three orthogonal axes: *provability*, *logical strength to informal*, and *root cause*. The first two axes describe *what* a wrong GT does to the evaluative claim a benchmark item is meant to support; the third describes *why* the error arose, and is what a benchmark builder or auditor can act on. We arrived at the root-cause categories by iterative refinement—seeding a small set from an initial reading of ProofNet, then splitting, merging, and adding categories across the full manual audit until the set stabilized and the *other* escape category went unused (every unfaithful item falls into a named cause). Table 1 summarizes the audit along all three axes.

Provability. Whether the formalized statement is mathematically true. If it is false, we should be able to prove its negation in Lean (Moura & Ullrich, 2021). A false GT is the most damaging case: as Section 3 shows, no sound theorem prover can produce a positive signal on it, and no faithful autoformalization can land on it. Of the 204 unfaithful items, 104 (51%) are mathematically false; of the remaining 100, 99 are mathematically true statements that simply differ from their NL source and 1 is of undetermined provability (the "Unknown" row of Table 1).

Logical strength to informal. Whether a candidate formalization is logically stronger than, weaker than, equivalent to, or incomparable with the intended (correct) formalization. Checking implications modulo the same background theory T, G is *equivalent* (hence faithful) when $G \Leftrightarrow I$, *stronger* when $G \Rightarrow I$ but $I \not\Rightarrow G$, *weaker* when $I \Rightarrow G$ but $G \not\Rightarrow I$, and *incomparable* when neither implication holds. A GT can look incomparable in surface form yet still be equivalent modulo T; we resolve such cases to *faithful* rather than logging them as errors. This axis is what equivalence-based autoformalization metrics try to detect, and it determines whether a theorem prover's verdict over- or under-estimates capability. Of the 194 unfaithful items where a uniquely correct GT is well-defined, 75 are strictly stronger than the correct GT, 55 strictly weaker, and 64 incomparable—roughly an even split across the three off-equivalence relations. A further 10 items admit no such comparison because the *benchmark item itself*, not the formalization, is at fault: the NL source either is *ambiguous*—admitting two or more mutually non-equivalent formal readings with no in-source disambiguator (9 items)—or is *mathematically incorrect*—the NL claim is itself false (1 item)—so it pins down no unique I to compare against.

Root cause. The translation mistake that produced the wrong GT. The concrete error categories enumerated below (and in the right panel of Table 1) *are* the values of this third axis: provability and logical strength describe the downstream *consequences* of an error, while root cause names the mistake itself. Each cause has a characteristic effect on the first two axes; this mapping from cause to logical strength to evaluative bias lets the taxonomy predict how a given error class will distort downstream evaluation.

Misunderstanding of informal math (94 / 204). The formalization misreads the NL source itself, before any Lean is involved. In Figure 1, S^1 (the unit circle in $\mathbb{R}^2$) is misread as the unit sphere of $\mathbb{R}$, collapsing the domain to two points. These errors typically produce a *strictly weaker* formalization and are the modal cause, $3\times$ the next category.

Missing premise (30 / 204). The NL source explicitly states a hypothesis that the formalization drops (Figure 7). The result is a *strictly stronger* formalization, frequently *false*.

Misunderstanding of Mathlib (26 / 204). The formalization uses a Mathlib (The mathlib Community, 2020) definition with semantics that diverge from informal usage—e.g., `Monotone` denotes non-decreasing rather than monotonic in either direction (Figure 2). These mismatches flip the formalization to a strictly stronger or strictly weaker statement depending on the direction of divergence.

Implicit premise (18 / 204). The formalization omits a hypothesis required for the NL claim but not stated explicitly in the source—a common pitfall in textbook prose, where authors rely on conventions a literal translator does not infer. Like a missing premise, this yields a *stronger* and often *false* formalization (Figure 8: omitting connectedness of Ω makes the claim false on disconnected domains). To separate this case from an unfaithful strengthening, the reviewer applies

Provability
(204 unfaithful)

	count	share
False	104	51.0%
True	99	48.5%
Unknown	1	0.5%

Faithfulness verdict
(all 367 items)

	count	share
Faithful	163	44.4%
Stronger than GT	75	20.4%
Weaker than GT	55	15.0%
Incomparable to GT	64	17.4%
NL ambiguous	9	2.5%
NL incorrect	1	0.3%

Root cause
(204 unfaithful items)

	count	share
Misunderstanding of informal math	94	46.1%
Missing premise	30	14.7%
Misunderstanding of Mathlib	26	12.7%
Implicit premise	18	8.8%
Misunderstanding of Lean	10	4.9%
Adding extra case	9	4.4%
Adding extra conclusion	7	3.4%
NL source ambiguous	9	4.4%
NL source incorrect	1	0.5%

Table 1. ProofNet audit: 204/367 items (56%) are unfaithful; 104 of those are mathematically false.

Misunderstanding of Informal Math

Natural Language

Let $f : S^1 \to \mathbb{R}$ be a continuous map. Show there exists a point x of S^1 such that $f(x) = f(-x)$.

Incorrect Formalization

```
theorem Munkres_exercise_24_2
  {f : (Metric.sphere 0 1 : Set ℝ) →
    ℝ}
  (hf : Continuous f) :
  ∃ x, f x = f (-x)
```

Ground Truth

```
theorem Munkres_exercise_24_2
  {f : (Metric.sphere (0 : ℂ) 1
        : Set ℂ) → ℝ}
  (hf : Continuous f) :
  ∃ x, f x = f (-x)
```

Figure 1. **Misunderstanding of informal math.** S^1 is misread as a sphere in $\mathbb{R}$, collapsing the domain to $\{-1, 1\}$.

Misunderstanding of Mathlib

Natural Language

Prove that every continuous open mapping of $\mathbb{R}^1$ into $\mathbb{R}^1$ is monotonic.

Incorrect Formalization

```
theorem Rudin_exercise_4_15
  {f : ℝ → ℝ}
  (hf : Continuous f)
  (hof : IsOpenMap f) :
  Monotone f
```

Ground Truth

```
theorem Rudin_exercise_4_15
  {f : ℝ → ℝ}
  (hf : Continuous f)
  (hof : IsOpenMap f) :
  Monotone f ∨ Antitone f
```

Figure 2. **Misunderstanding of Mathlib.** Mathlib's `Monotone` means non-decreasing only; the correct statement requires the disjunction with `Antitone`.

a single policy: a hypothesis is an *implicit premise* (and recovering it is faithful) only when the NL source presupposes it—by convention, context, or the standard reading a domain expert would supply—so the intended claim I already contains it; a hypothesis the source does not presuppose instead *narrows* the claimed generality and is logged as an unfaithful strengthening. The certifying Lean proof disciplines this: the recovered premise must be exactly what is needed to make the intended claim provable, no stronger.

Misunderstanding of Lean (10 / 204). The formalization misuses Lean's type system or syntax independently of any specific Mathlib API—e.g., expressing "let R be the quotient ring" as a propositional type equality `R = A / I` between bare `Type`s (Figure 9). These errors often render the hypothesis vacuous or the conclusion trivial, producing an *easier* GT that is incomparable to the intended one.

These five causes together account for 178 of 204 unfaithful

items (87%). The remaining 26 split across three smaller patterns: **additive specification errors** (16 items: 9 extra-case disjunctions and 7 extra-conjunct conclusions, structurally mirroring missing premises and yielding strictly stronger and often false formalizations); **NL ambiguous** (9 items where the NL source admits multiple incompatible formal readings, e.g. "regular" as `RegularSpace` vs. `CompletelyRegularSpace`, making any equivalence-based metric report interpretive disagreement rather than model capability; Figure 10); and **NL incorrect** (1 item whose NL source is itself mathematically wrong).

3. Analysis

3.1. Qualitative

A wrong ground truth (GT) corrupts evaluation in different ways depending on the task. For theorem proving, a sound prover never lies, so the corruption shows up as biased pass/-fail rates relative to true model ability. For autoformalization, equivalence-based judgments compress the measured gap between models through two compounding mechanisms (capping and coincidental agreement), with stronger models penalized *more* than weaker ones.

Theorem proving. For each benchmark item, a prover either returns a Lean proof of the GT formalization or fails. Figure 3 enumerates the eight possible outcomes, organized by the prover's verdict (rows) and the GT's relationship to the underlying natural-language (NL) claim (columns).

When the GT is *correct*, pass/fail directly reflects ability. The other three columns each leak signal differently. **GT easier:** a prover can pass because the formalization is weaker than the NL claim, scoring a problem the model could not actually solve and over-estimating capability (though a failure is still informative, since a model failing the easier statement would also fail the intended one). **GT false:** no sound prover can prove a false statement, so the only achievable verdict is failure—capping the measurable signal, inflating apparent benchmark difficulty, and contributing zero discriminative power between models. **GT harder:** a prover that succeeds has shown something strictly stronger than the NL claim, which we still count correct; but failure does not imply the model could not have proved the intended claim, so we under-estimate capability. The "easier" and "harder" columns bias scores in opposite directions and need not cancel: the rates of each error type are not balanced in the benchmarks we audit.

Autoformalization. Autoformalization is evaluated by checking whether the model's predicted formalization is equivalent to the GT formalization. When the GT is faithful, this is a sound test. When the GT is wrong, equivalence checking can systematically compress what we are trying to measure, as illustrated in Figure 4.

Wrong GTs compress measured gaps through two compounding mechanisms (Figure 4). (i) **Capping**: the NL input is equivalent by construction to the correct GT, so any faithful translation—strong or weak—is judged non-equivalent to a differing wrong GT; the unfaithful subset thus collapses near zero across the board, contributing little discriminative signal. (ii) **Coincidental agreement**, which is asymmetric: a weak model's prediction may match the wrong GT by coincidence, especially when that GT reflects the same misreading a noisy translator is likely to reproduce, shifting a small fraction of unfaithful items toward rewarding the weak model. The consequence is counter-intuitive—**equivalence-based metrics on a wrong GT penalize models for being faithful**—and, aggregated over a benchmark with a non-trivial fraction of wrong GTs, compresses the apparent quality gap between models. Empirically on ProofNet (Section 3.2), capping accounts for $\sim$5 pp of the $\sim$6 pp aggregate compression and coincidental agreement for $\sim$1 pp.

3.2. Quantitative

To confirm the analysis empirically, we evaluate two open-weight models on ProofNet using an LLM-based theorem prover with up to five rounds of Lean compilation feedback per problem: **Qwen3.5-27B** and **Qwen3.5-397B-A17B**, both in *thinking* mode. The two share an architecture and training recipe but differ by $\sim 15\times$ in scale, isolating capability as the only varying factor: any change in their measured gap from cleaning the benchmark is attributable to the ground truth rather than to model family, tokenizer, or prompting. This controlled size pair exposes how dirty GTs distort the *gap between* models more cleanly than a heterogeneous set of provers would; the autoformalization study below adds a stronger closed model (claude-opus-4.6) to confirm the cross-family order is preserved. We measure pass rate on (i) the original ProofNet release ("OG") and (ii) ProofNet-Verified.

Cleaning widens the absolute gap but preserves the ratio. On the original ProofNet, 27B passes 9.5% of items and 397B 13.6%. On ProofNet-Verified, the rates rise to $\sim 12\%$ and $\sim 17\%$: both models gain roughly 3 pp, the multiplicative gap stays at $\sim 1.4\times$, and the absolute gap widens from $\sim$4 pp to $\sim$5 pp—the GT-false capping effect made concrete.

Per-verdict pass rates trace the qualitative prediction. Table 2 decomposes pass rate by faithfulness verdict. *Faithful* items pass at the rate reflecting true capability, with clear separation between sizes (14.7% vs 19.0%); *stronger* items collapse to single digits (5.3% for both) because most are false; and *incomparable* items pass $0/64$. This floor

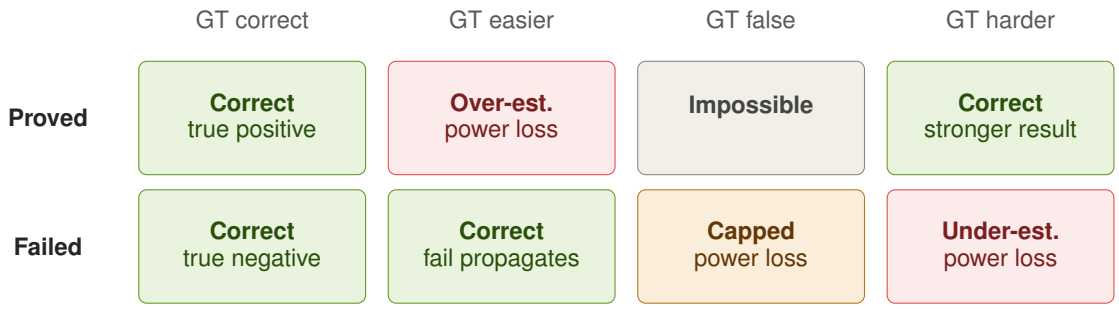

Figure 3. Effect of wrong GTs on theorem-proving evaluation. "GT easier" over-estimates capability; "GT false" caps signal; "GT harder" under-estimates capability.

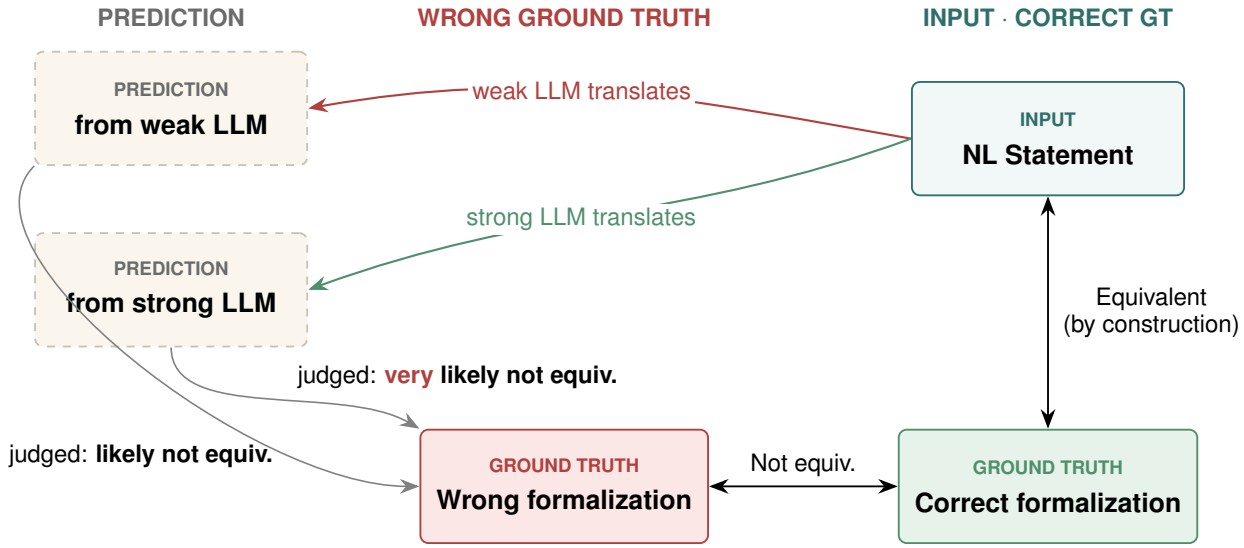

Figure 4. Gap-compression mechanism: a faithful (strong) model is rejected against a wrong GT, while a weak model may agree by coincidence. Aggregated, this compresses the measured capability gap between models.

is structural: 46 of 64 items are false, and the remaining 17 provable ones formalize a different claim than the NL suggests, giving the prover no semantic anchor. The *weaker* row holds only for the larger model: 397B passes 20.0% (slightly above its faithful rate) while 27B drops to 9.1%, suggesting capability rather than slack binds the smaller model.

Autoformalization: cleaning the benchmark restores the model gap. The qualitative prediction (Section 3.1, Figure 4) is that equivalence-based metrics on a wrong GT *compress* the apparent gap between models. We test this with BEq+ on six autoformalizer configurations—claude-opus-4.6 at two effort levels, and two qwen3.5 sizes (27B and 397B-A17B) each with and without thinking—over the same 353 ProofNet items, scored first against the original ground truths and then against ProofNet-Verified. Table 3 reports the headline result: every configuration gains 7.1–14.1 percentage points from cleaning, and the gap between the strongest and weakest configuration widens from 4.0 pp on ProofNet to 9.9 pp on ProofNet-Verified—a 60% compression of the apparent capability difference under dirty GTs.

Compression at its limit produces rank inversion. Within the qwen3.5-397B family, compression produces a literal rank flip: on ProofNet, no-thinking scores 19.3% and thinking 18.4%—the metric falsely reports that thinking *hurts*—while on ProofNet-Verified the order reverses (30.6% vs 27.2%), recovering the signal that thinking helps. We do not generalize from a single within-family flip (the cross-family order Claude > Qwen is preserved on both datasets), but it is the cleanest rendering of Figure 4: under dirty GTs an intervention with a strong prior can flip into a directionally-incorrect measurement. Figure 5 shows the same mechanism for a single item.

Restricting to the corrected subset reveals aggregate rank inversion. On the 180 unfaithful items (stronger/weaker/incomparable), Table 4 shows claude-opus-4.6 high passing 1.1% on ProofNet vs. qwen3.5-27B no-thinking's 2.2%—the strongest configuration outranked

		27B		397B	
Faithfulness verdict	n	ProofNet	ProofNet-Verified	ProofNet	ProofNet-Verified
faithful	163	14.7%	14.7%	19.0%	19.0%
weaker	55	9.1%	9.1%	20.0%	20.0%
stronger	75	5.3%	15.8%	5.3%	10.5%
incomparable	64	0.0%	0.0%	1.6%	0.0%
nl ambiguous	9	22.2%	25.0%	33.3%	37.5%
nl incorrect	1	0.0%	0.0%	0.0%	0.0%
overall	367	9.5%	12.2%	13.6%	17.4%

Table 2. Pass rates by faithfulness verdict for two Qwen3.5 model sizes (thinking mode), measured on Original and projected onto ProofNet-Verified. Verified figures use each model's pass rate on the *provable* subset within each category as a proxy for its capability after correction; for *faithful* and *weaker* all items are provable in ProofNet, so they're equivalent. The largest cleaning gains land on *stronger*.

Table 3. **Cleaning the benchmark restores the apparent model gap.** BEq+ autoformalization pass rates on the same 353 ProofNet items, evaluated against the original ProofNet ground truths and against the corrected ProofNet-Verified ground truths. All six model/thinking configurations gain 7.9–14.1 percentage points on ProofNet-Verified, but the gain is uneven across model strength: the strongest-minus-weakest gap (claude-opus-4.6 high vs qwen3.5-27B no-thinking) is 4.0 pp on ProofNet but 9.9 pp on ProofNet-Verified, a 60% compression of the apparent capability difference under dirty GTs. The pattern replicates with the 397B-vs-nothink comparison (5.4 pp compression) and holds across three model sizes (27B, 397B, claude-opus-4.6), confirming the gap-compression mechanism of Section 3.1 made concrete: on unfaithful items both models collapse near zero (capping), with a small additional coincidental-agreement effect, so cleaner GTs separate models rather than compress them—and within the qwen3.5-397B family the compression is large enough to produce a literal rank flip between thinking and no-thinking. 14 items dropped from both datasets because their original GT is declared as an `instance` rather than a `theorem`.

Autoformalizer	ProofNet	ProofNet-Verified	Cleaning lift
claude-opus-4.6 (high effort)	22.4%	35.7%	+13.3 pp
claude-opus-4.6 (low effort)	21.0%	35.1%	+14.1 pp
qwen3.5-397B-A17B (thinking)	18.4%	30.6%	+12.2 pp
qwen3.5-397B-A17B (no thinking)	19.3%	27.2%	+ 7.9 pp
qwen3.5-27B (thinking)	18.7%	25.8%	+ 7.1 pp
qwen3.5-27B (no thinking)	18.4%	25.8%	+ 7.4 pp
Strongest − weakest gap	+4.0 pp	+9.9 pp	—

by the weakest in aggregate, not just within-family. Cleaning restores the order (25.6% vs. 16.7%, a +8.9 pp gap), confirming that the aggregate compression on all 353 items is driven entirely by the unfaithful subset, diluted by the unchanged faithful items.

Per-verdict AF rates decompose the compression cell by cell. On ProofNet, every configuration shows true capability *only* on the 163 faithful items (38–47%); the 180 unfaithful items collapse to 0–4% across the board, contributing essentially no discriminative signal—the capping mechanism. Coincidental agreement is visible on the *stronger* row where the weakest configuration outperforms the strongest by 2.8 pp. Cleaning lifts the three logically-related unfaithful classes by 7–31 pp, bringing them within reach of the faithful rate; the *incomparable* subset remains the hardest (7–16 pp lift), the same failure mode that capped theorem proving. Full per-verdict table in Appendix B.

4. Auditing Pipeline

The audit results in Section 2 were produced by a semi-automated pipeline that takes a benchmark item q—a (natural language, formal ground truth) pair—and emits a verdict on whether the GT faithfully encodes the NL claim, a taxonomy label, and, when needed, a corrected formalization. Running this pipeline on all 367 items of ProofNet produces **ProofNet-Verified**: a corrected version of the benchmark in which every item is either the original GT certified by the audit or a revision that is. All provability checks use **Lean v4.28.0** with the matching Mathlib release; ProofNet-Verified ports the original Lean 3 ProofNet to this toolchain via the Lean 4 ProofNet# release of Poiroux et al. (2025), so our corrections and proofs target current Mathlib APIs. Figure 6 illustrates the four stages.

First, we attempt to prove both q and $\neg q$ in Lean with an agentic LLM prover (Claude Opus 4.6) that receives Lean kernel feedback and iteratively repairs its proof, with web

Table 4. **Restricting to the 180 corrected items reveals aggregate rank inversion.** BEq+ pass rates on the subset of ProofNet items where the GT was unfaithful (stronger / weaker / incomparable in our audit). On the dirty GTs the strongest configuration is outranked by the weakest in aggregate—not just within-family—and cleaning restores the directionally-correct ordering. The ∼10 pp swing on this subset is consistent with the all-353 compression in Table 3 being driven entirely by the unfaithful items, diluted by the 163 faithful items where the GT is unchanged.

Autoformalizer	ProofNet	ProofNet-Verified	Cleaning lift
claude-opus-4.6 (high effort)	1.1%	25.6%	+24.5 pp
claude-opus-4.6 (low effort)	2.2%	25.0%	+22.8 pp
qwen3.5-397B-A17B (thinking)	1.1%	22.8%	+21.7 pp
qwen3.5-397B-A17B (no thinking)	0.6%	17.7%	+17.2 pp
qwen3.5-27B (thinking)	2.2%	15.6%	+13.4 pp
qwen3.5-27B (no thinking)	2.2%	16.7%	+14.5 pp
Strongest − weakest gap	−1.1 pp	+8.9 pp	—

Worked example: coincidental agreement on Herstein 4.3.25

Natural Language

Let R be the ring of 2×2 matrices over the reals; suppose that I is an ideal of R. Show that $I = (0)$ or $I = R$.

Standard mathematical reading. "Ideal of a non-commutative ring" means *two-sided* ideal; the claim is true and is a special case of Wedderburn–Artin.

ProofNet GT (Mathlib `Ideal`)

```
theorem Herstein_4_3_25
  (I : Ideal (Matrix (Fin 2)
              (Fin 2) ℝ)) :
  I = ⊥ ∨ I = ⊤
```

Mathematically false: $M_2(\mathbb{R})$ has non-trivial left ideals (e.g. matrices with zero second column).

ProofNet-Verified GT (`TwoSidedIdeal`)

```
theorem Herstein_4_3_25
  (I : TwoSidedIdeal
       (Matrix (Fin 2)
        (Fin 2) ℝ)) :
  I = ⊥ ∨ I = ⊤
```

Mathematically true.

Two autoformalizers, same input:

- **claude-opus-4.6 (strong)** predicts `TwoSidedIdeal`—faithful to NL, agrees with ProofNet-Verified, *disagrees with ProofNet*. BEq+ judges *not equivalent* against ProofNet. **Penalized.**
- **qwen3.5-27B (weak)** predicts `Ideal`—reproduces the same Mathlib mistake the ProofNet author made, agrees with ProofNet, *disagrees with ProofNet-Verified*. BEq+ judges *equivalent* against ProofNet. **Rewarded.**

On ProofNet, this item contributes a point to the weaker model and a zero to the stronger one—the coincidental-agreement mechanism of Section 3.1 made concrete on a single benchmark item, with the weak model's verdict landing above the strong model's.

Figure 5. **Verdict flip in action.** On Herstein 4.3.25, claude-opus-4.6 produces the faithful formalization (`TwoSidedIdeal`) and is judged non-equivalent against the wrong ProofNet GT, while qwen3.5-27B reproduces the GT author's `Ideal` mistake and is judged equivalent. On ProofNet-Verified the verdicts flip—the coincidental-agreement mechanism (Section 3.1, Figure 4) on a single item.

search over Mathlib enabled and a hard constraint forbidding `sorry`, `admit`, and new axioms. A machine-checked proof of $\neg q$ is a hard refutation that routes the item to correction, while a proof of q alone is only weakly informative since a true formalization can still be unfaithful. The benchmark is thus certified end-to-end: every one of the 367 shipped statements carries a complete, `sorry`-free Lean proof, and every false original is accompanied by a machine-checked refutation ($\neg q$), all checked by the Lean kernel. Second, for each refuted item, we diagnose the defect and prove a revised statement under the same constraints. Third, each candidate (original or revised) is checked for faithfulness by an LLM judge run three independent times; the verdict is stochastic, so we require unanimity—the item passes only when all three return zero issues, and any single dissent routes it to human review rather than silently passing.

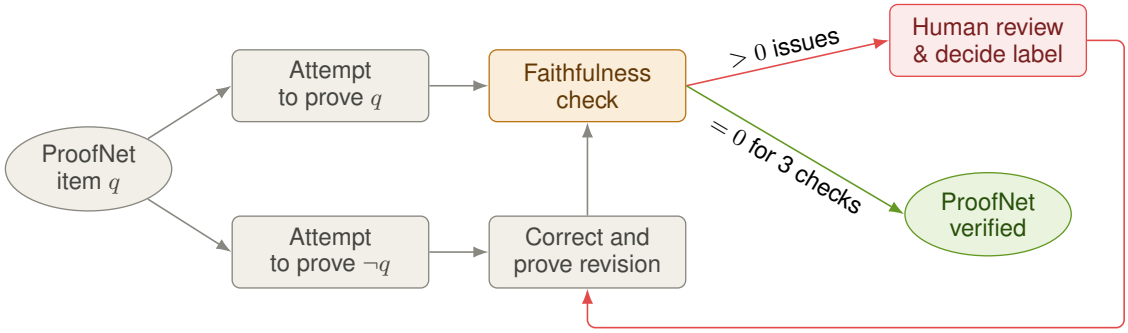

Figure 6. Semi-automated audit pipeline. Provability checks, LLM faithfulness judge ($\times 3$), and human review produce a verified or labeled item.

Fourth, a human reviewer audits the judge's proposed labels for every item and commits the final ones, either confirming a taxonomy label (e.g., *NL ambiguous*) or proposing an edit that re-enters the pipeline. This audit caught at least 11 LLM-judge errors out of 367: 4 items the judge wrongly accepted as *faithful* (reclassified as unfaithful), 5 further items mis-ranked on the stronger/weaker/incomparable axis, and 2 provability verdicts the judge had left undetermined that a Lean proof or refutation then resolved. The judge thus agreed with the final human label on $363/367$ (98.9%) of faithful-vs-unfaithful decisions and $358/367$ (97.5%) of full logical-strength verdicts—a strong but imperfect filter, which is why human review is the authoritative final stage.

The pipeline produces two artifacts. **ProofNet-Verified** replaces every unfaithful GT with either a revised, audit-certified formalization (where a clean revision exists) or a flag indicating the item is unsuitable for equivalence-based evaluation (the 10 items where the NL source is itself ambiguous or incorrect). Alongside the dataset, every one of the 367 items is annotated with a per-item taxonomy label—*faithful* for the 163 items that passed unmodified, or, when not, the root cause together with the logical-strength and provability annotations underlying Table 1. The same pipeline is what we apply in Section 5 to PutnamBench, FATE, ProverBench, and CombiBench.

5. Estimation

To test whether the patterns observed on ProofNet are idiosyncratic or general, we applied the audit pipeline of Section 4 to six other formal-math benchmarks: **PutnamBench** (Tsoukalas et al., 2024), **ProverBench** (Ren et al., 2025), **CombiBench** (Liu et al., 2025), **LeanCat** (Xu et al., 2025), and the **FATE** family (Jiang et al., 2026) (M, H, X). For ProofNet we ran the full pipeline, including human review, yielding a *confirmed* unfaithful count; for the remaining benchmarks we ran only the automated stages (provability checks + LLM faithfulness judge) due to scale, yielding *estimated* counts. We stress that these six rates are LLM-judge estimates without the human adjudication that backs the ProofNet figure, and should be read as approximate rather than certified: they inherit the judge's error rate, so individual per-benchmark numbers may move under full human review even though the order-of-magnitude spread is robust. Table 5 reports the headline rates; per-benchmark verdict breakdowns are deferred to Appendix D.

Benchmark	Problems	Correct	Wrong	Error %
PutnamBench	672	640	32	4.8%
FATE-H	100	95	5	5.0%
FATE-M	150	135	15	10.0%
FATE-X	100	86	14	14.0%
FATE (total)	**350**	**316**	**34**	**9.7%**
LeanCat	100	83	17	17.0%
CombiBench	100	72	28	28.0%
ProofNet†	367	163	204	55.6%
ProverBench	325	130	195	60.0%

Table 5. Cross-benchmark audit summary, sorted by error rate. ProofNet (†) used the full pipeline with human review; others used the LLM judge only.

Unfaithful-GT rates span more than an order of magnitude. PutnamBench is the cleanest benchmark we audit, with only 32 of 672 items (4.8%) flagged as unfaithful. At the other end, ProverBench has an estimated 60.0% unfaithful rate, on par with ProofNet (55.6%). The FATE family clusters at the cleaner end (5–14% per split), with FATE-H the cleanest at 5% and FATE-X the most corrupted within the family at 14%; CombiBench (28%) and LeanCat (17%) sit in the middle.

Patterns generalize; absolute rates do not. Every benchmark exhibits the logical-strength failure modes catalogued in Section 2—stronger, weaker, and incomparable formalizations—so the taxonomy itself transfers without modification. CombiBench surfaces one additional failure mode not seen in ProofNet: 6 of 100 items are *uncompilable*, meaning the released GT does not type-check against the declared Lean version, defeating evaluation outright. What

does not transfer between benchmarks is the rate: a $12\times$ spread (4.8%–60.0%) across these benchmarks shows that ground-truth quality is dominated by curatorial process, not by source material or task difficulty. PutnamBench's near-cleanliness is an existence proof that low-error benchmarks are achievable; the rest of the field is far from that ceiling.

6. Related Work

Auditing benchmark integrity. Label errors in ML benchmarks distort model rankings (Beyer et al., 2020; Northcutt et al., 2021); recent audits of MMLU (Gema et al., 2025) and other NLP test sets (Nahum et al., 2025) confirm this for language models, while Vendrow et al. (2025) and Tu et al. (2026) address benchmark reliability and automated auditing more broadly. We extend this tradition to formal-math benchmarks, where the ground truth is a Lean statement rather than a class label, offering a sharper handle: a proof of $\neg q$ is a machine-checked refutation, and the logical structure of formal statements lets us classify errors by their effect on evaluation (Section 2).

Formal-math benchmarks and autoformalization evaluation. The benchmarks we audit are all formalized in Lean (Moura & Ullrich, 2021) against Mathlib (The mathlib Community, 2020), spanning undergraduate exercises (ProofNet (Azerbayev et al., 2023)), competition mathematics (PutnamBench (Tsoukalas et al., 2024), CombiBench (Liu et al., 2025)), frontier algebra (FATE (Jiang et al., 2026)), category theory (LeanCat (Xu et al., 2025)), and AIME/textbook problems (ProverBench (Ren et al., 2025)). On the autoformalization-evaluation side, Poiroux et al. (2025) use Lean's type checker to filter ill-formed candidate translations, reducing reliance on equivalence-based judges; that effort also manually repaired prior Lean 4 ProofNet formalizations and released ProofNet# (the base our port builds on). Our work is complementary: rather than improving the equivalence judge or its inputs, we audit the ground truth it is judged against, and rather than manual correction alone we certify each statement with a machine-checked proof, classify every error by a taxonomy, and quantify the downstream evaluation bias. Closest in spirit but on a different benchmark, Ospanov et al. (2025) revisit miniF2F, cataloguing its limitations and proposing a lighter-weight taxonomy; our three-axis taxonomy is finer-grained and tied directly to predictions about evaluation bias.

7. Conclusion

Our audit shows that 56% of ProofNet ground truths are unfaithful—over half mathematically false—biasing evaluation by capping theorem-proving signal and compressing autoformalization gaps. We release **ProofNet-Verified** with per-item taxonomy labels for all 367 items, and the same pipeline surfaces comparable patterns across six other benchmarks. Ground-truth verification should be a routine step in releasing and consuming formal-math benchmarks, not an exceptional one.

Limitations. Only ProofNet received the full pipeline with human review; the other six benchmarks have LLM-judge estimates ($\sim$10% error). Our audit covers only Lean/Mathlib; whether the patterns hold in Coq or Isabelle is untested. The theorem-proving evaluation (Section 3.2) uses two models on single runs.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgments

This work is supported by an AWS ASSET Ph.D. Fellowship, the ProML project funded by the Canadian Natural Sciences and Engineering Research Council and the French National Research Agency under the reference ANR-25-CE23-6715, NSF Award CCF-2338777, Amazon Research Award Fall 2023, Amazon/ASSET Gift for Research in Trustworthy AI. Any opinions, findings, conclusions, or recommendations expressed herein are those of the authors and do not necessarily reflect the views of funding entities.

References

Azerbayev, Z., Piotrowski, B., Schoelkopf, H., Ayers, E. W., Radev, D., and Avigad, J. Proofnet: Autoformalizing and formally proving undergraduate-level mathematics, 2023. URL https://arxiv.org/abs/2302.12433.

Beyer, L., Hénaff, O. J., Kolesnikov, A., Zhai, X., and van den Oord, A. Are we done with imagenet?, 2020. URL https://arxiv.org/abs/2006.07159.

Gema, A. P., Leang, J. O. J., Hong, G., Devoto, A., Mancino, A. C. M., Saxena, R., He, X., Zhao, Y., Du, X., Madani, M. R. G., Barale, C., McHardy, R., Harris, J., Kaddour, J., van Krieken, E., and Minervini, P. Are we done with mmlu?, 2025. URL https://arxiv.org/abs/2406.04127.

Jiang, J., He, W., Wang, Y., Gao, G., Hu, Y., Wang, J., Guan, N., Wu, P., Dai, C., Xiao, L., and Dong, B. Fate: A formal benchmark series for frontier algebra of multiple difficulty levels, 2026. URL https://arxiv.org/abs/2511.02872.

Liu, J., Lin, X., Bayer, J., Dillies, Y., Jiang, W., Liang, X., Soletskyi, R., Wang, H., Xie, Y., Xiong, B., Yang, Z., Zhang, J., Zhi, L., Li, J., and Liu, Z. Combibench: Benchmarking llm capability for combinatorial mathematics, 2025. URL https://arxiv.org/abs/2505.03171.

Moura, L. d. and Ullrich, S. The Lean 4 theorem prover and programming language. In Platzer, A. and Sutcliffe, G. (eds.), *Automated Deduction – CADE 28*, pp. 625–635, Cham, 2021. Springer International Publishing. ISBN 978-3-030-79876-5.

Nahum, O., Calderon, N., Keller, O., Szpektor, I., and Reichart, R. Are llms better than reported? detecting label errors and mitigating their effect on model performance, 2025. URL https://arxiv.org/abs/2410.18889.

Northcutt, C. G., Athalye, A., and Mueller, J. Pervasive label errors in test sets destabilize machine learning benchmarks, 2021. URL https://arxiv.org/abs/2103.14749.

Ospanov, A., Farnia, F., and Yousefzadeh, R. minif2f-lean revisited: Reviewing limitations and charting a path forward, 2025. URL https://arxiv.org/abs/2511.03108.

Poiroux, A., Weiss, G., Kunčak, V., and Bosselut, A. Reliable evaluation and benchmarks for statement autoformalization, 2025. URL https://arxiv.org/abs/2406.07222.

Ren, Z. Z., Shao, Z., Song, J., Xin, H., Wang, H., Zhao, W., Zhang, L., Fu, Z., Zhu, Q., Yang, D., Wu, Z. F., Gou, Z., Ma, S., Tang, H., Liu, Y., Gao, W., Guo, D., and Ruan, C. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition, 2025. URL https://arxiv.org/abs/2504.21801.

The mathlib Community. The Lean Mathematical Library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2020, New Orleans, LA, USA, January 2020. ACM. doi: 10.1145/3372885.3373824. URL https://doi.org/10.1145/3372885.3373824.

Tsoukalas, G., Lee, J., Jennings, J., Xin, J., Ding, M., Jennings, M., Thakur, A., and Chaudhuri, S. Putnambench: Evaluating neural theorem-provers on the putnam mathematical competition, 2024. URL https://arxiv.org/abs/2407.11214.

Tu, X., Wang, T., Lu, Y., Huang, K., Qu, Y., and Mostafavi, S. Benchguard: Who guards the benchmarks? automated auditing of llm agent benchmarks, 2026. URL https://arxiv.org/abs/2604.24955.

Vendrow, J., Vendrow, E., Beery, S., and Madry, A. Do large language model benchmarks test reliability?, 2025. URL https://arxiv.org/abs/2502.03461.

Xu, R., Dai, H., Fu, Y., Jiang, J., Nie, T., Wang, J., Yang, H., and Zhang, Z.-H. Leancat: A benchmark suite for formal category theory in lean (part i: 1-categories), 2025. URL https://arxiv.org/abs/2512.24796.

Missing Premise

Natural Language

Let G be a finite group which possesses an automorphism σ such that $\sigma(g) = g$ if and only if $g = 1$. If σ^2 is the identity map from G to G, prove that G is abelian.

Incorrect Formalization

```
theorem Dummit_Foote_exercise_1_6_23
  {G : Type*} [Group G]
  (σ : MulAut G)
  (hs : ∀ g : G, σ g = g ↔ g = 1)
  (hs2 : ∀ g : G, σ (σ g) = g) :
  ∀ x y : G, x * y = y * x
```

Ground Truth

```
theorem Dummit_Foote_exercise_1_6_23
  {G : Type*} [Group G] [Finite G]
  (σ : MulAut G)
  (hs : ∀ g : G, σ g = g ↔ g = 1)
  (hs2 : ∀ g : G, σ (σ g) = g) :
  ∀ x y : G, x * y = y * x
```

Figure 7. **Missing premise.** The NL states G is finite, but the formalization drops `[Finite G]`.

Implicit Premise

Natural Language

Suppose that f is holomorphic in an open set Ω. Prove that if $\mathrm{Re}(f)$ is constant, then f is constant.

Incorrect Formalization

```
theorem Shakarchi_exercise_1_13a
  {f : ℂ → ℂ} (Ω : Set ℂ) (a b : Ω)
  (h : IsOpen Ω)
  (hf : DifferentiableOn ℂ f Ω)
  (hc : ∃ (c : ℝ), ∀ z ∈ Ω,
        (f z).re = c) :
  f a = f b
```

Ground Truth

```
theorem Shakarchi_exercise_1_13a
  {f : ℂ → ℂ} (Ω : Set ℂ) (a b : Ω)
  (h : IsOpen Ω)
  (hconn : IsPreconnected Ω)
  (hf : DifferentiableOn ℂ f Ω)
  (hc : ∃ (c : ℝ), ∀ z ∈ Ω,
        (f z).re = c) :
  f a = f b
```

Figure 8. **Implicit premise.** Connectedness of Ω is silently assumed in the NL but missing from the formalization, making it false.

A. Additional Error-Type Figures

Section 2 illustrates two root causes in the body—misunderstanding of informal math (Figure 1) and misunderstanding of Mathlib (Figure 2). The four figures below cover the remaining categories from the taxonomy, each contrasting the natural-language source, the unfaithful ProofNet formalization, and a corrected version. The first two (missing premise, implicit premise) account for 48 of the 204 unfaithful items; the last two (misunderstanding of Lean, NL ambiguous) capture smaller patterns referenced in the body.

B. Per-Verdict Autoformalization Pass Rates

Table 6 decomposes BEq+ autoformalization pass rates by faithfulness verdict on both ProofNet and ProofNet-Verified, supplementing the aggregate result in Table 3. On ProofNet, every configuration shows true capability *only* on the 163 faithful items (38–47%); the 180 unfaithful items collapse to 0–4%, contributing essentially no discriminative signal between models (the capping mechanism of Section 3.1). Cleaning lifts the *stronger*, *weaker*, and *incomparable* rows by 7–31 pp, bringing them within reach of the faithful rate; the *incomparable* row remains the hardest (7–16 pp lift), reflecting the same "different formal claim than NL suggests" failure mode that caps theorem-proving signal.

Misunderstanding of Lean

Natural Language

Prove that the quotient ring $\mathbb{Z}[i]/(1+i)$ is a field of order 2.

Incorrect Formalization

```
theorem Dummit_Foote_exercise_8_3_6a
  {R : Type} [Ring R]
  (hR : R = (GaussianInt /
        span ({⟨1, 1⟩} : Set GaussianInt)
    )) :
  IsField R ∧
  ∃ finR : Fintype R,
  @card R finR = 2 := by
```

Ground Truth

```
theorem Dummit_Foote_exercise_8_3_6a :
  IsField (GaussianInt /
    span ({⟨1, 1⟩} : Set GaussianInt)) ∧
  ∃ finR : Fintype (GaussianInt /
    span ({⟨1, 1⟩} : Set GaussianInt)),
  @Fintype.card _ finR = 2
```

Figure 9. **Misunderstanding of Lean.** "Let R be the quotient ring" is expressed as a type equality `R = ...`, which is vacuous in Lean's type system.

NL Ambiguous

Natural Language

Show that if $\prod X_\alpha$ is regular, then so is X_α. Assume that each X_α is nonempty.

Formalization (one valid reading)

```
theorem Munkres_exercise_32_2b
  {ι : Type*} {X : ι → Type*} [∀ i, TopologicalSpace (X i)]
  (h : ∀ i, Nonempty (X i))
  (h2 : RegularSpace (Π i, X i)) :
  ∀ i, RegularSpace (X i)
```

Figure 10. **NL ambiguous.** "Regular" admits two Mathlib readings (`RegularSpace` vs. `CompletelyRegularSpace`); the NL alone cannot disambiguate.

C. Root-Cause Error Types of Autoformalizer Outputs

The audit in Section 2 classifies the *human* ProofNet ground truths; we apply the identical pipeline to the *autoformalizer* outputs to ask whether models reproduce the same error patterns. Table 7 compares the human ground truth against two autoformalizers—claude-opus-4.6 (high effort) and qwen3.5-397B (thinking)—each formalizing from the original (ProofNet) and corrected (ProofNet-V) informal prompts, with every candidate judged against ProofNet-Verified gold by the same LLM auditor used for Table 1.

Three findings stand out. First, faithfulness rises monotonically with model strength: the human formalizers are faithful only 44% of the time, qwen3.5-397B reaches ∼72%, and claude-opus-4.6 reaches ∼86%—both models far exceed the human rate, and each unfaithful tail is correspondingly smaller. Second, the *shape* of the error distribution is model-dependent rather than universal. The strong model echoes the human profile—*misunderstanding of informal math* is the modal cause (28–31%)—but adds a distinctive *adding extra case* over-specification (∼24–26% of its errors versus 4% for humans), almost all *weaker* and provably *true*, where it defensively widens the statement with an unrequested disjunct. The weaker model instead fails overwhelmingly through *misunderstanding of Mathlib* (∼52%, roughly 4× the human and strong-model rates): its bottleneck is library/API competence, not reading the mathematics. Third, error *severity* tracks strength too: only 16–17% of the strong model's errors are mathematically false (the rest weaker-but-true), against 51% for the human ground truth, whereas the weaker model regresses toward the human regime, with 58–64% of its errors false or unjudgeable. The faithfulness rates here reflect the LLM judge rather than BEq+, and are accordingly far higher than the equivalence-based pass rates in Table 6.

Table 6. **BEq+ pass rates by faithfulness verdict, on the original ProofNet and corrected ProofNet-Verified ground truths.** On ProofNet, each model's true AF capability is visible only on the 163 faithful items (36–47%); on the 180 unfaithful items pass rates collapse to 0–4% across every model and configuration, because the model correctly translates the natural-language claim and BEq+ then rejects it against the wrong GT. Cleaning lifts the three unfaithful classes by 8–30 pp, bringing them close to the faithful rate—direct empirical confirmation of Figure 4: equivalence-based AF metrics on unfaithful GTs penalize faithful models. The NL-AMBIGUOUS row is unstable because $n = 9$. Three autoformalizers $\times$ two reasoning levels each, same 353-item base as Table 3; column abbreviations: C-H = claude-opus-4.6 high, C-L = claude-opus-4.6 low, Q397-T / Q397-N = qwen3.5-397B thinking / no-thinking, Q27-T / Q27-N = qwen3.5-27B thinking / no-thinking.

(a) ProofNet (original ground truths)

Verdict	n	C-H	C-L	Q397-T	Q397-N	Q27-T	Q27-N
faithful	163	46.6	42.3	38.0	39.3	36.8	36.2
stronger	71	0.0	2.8	1.4	1.4	2.8	2.8
weaker	47	2.1	4.3	0.0	0.0	2.1	2.1
incomparable	62	1.6	0.0	1.6	0.0	1.6	1.6
nl ambiguous	9	11.1	11.1	11.1	33.3	22.2	22.2
overall	353	22.4	21.0	18.4	19.3	18.7	18.4

(b) ProofNet-Verified (corrected ground truths)

Verdict	n	C-H	C-L	Q397-T	Q397-N	Q27-T	Q27-N
faithful	163	46.0	47.2	40.5	37.4	36.8	35.6
stronger	71	29.6	33.8	31.0	22.5	16.9	19.7
weaker	47	31.9	23.4	23.4	19.1	19.1	23.4
incomparable	62	16.1	16.1	12.9	11.3	11.3	8.1
nl ambiguous	9	44.4	11.1	11.1	22.2	33.3	33.3
overall	353	35.7	35.1	30.6	27.2	25.8	25.8

D. Per-benchmark Faithfulness Breakdowns

For each non-ProofNet benchmark we audit, Table 8 reports the verdict distribution as estimated by the LLM faithfulness judge. These are the same data summarized in Table 5, broken out by logical-strength verdict.

	ProofNet	**claude-opus-4.6** (high)		**qwen3.5-397B** (think)	
	human GT	ProofNet	ProofNet-V	ProofNet	ProofNet-V
	n=367	353	367	353	367
Faithfulness verdict (% of all items)					
Faithful	44.4	85.6	87.2	73.4	71.1
Stronger than GT	20.4	4.0	4.1	4.5	4.4
Weaker than GT	15.0	6.5	6.3	5.1	7.4
Incomparable to GT	17.4	4.0	2.5	17.0	17.2
NL ambiguous	2.5	0.0	0.0	0.0	0.0
NL incorrect	0.3	0.0	0.0	0.0	0.0
Root cause (% of unfaithful; $n = 204/51/47/94/106$)					
Misunderstanding of informal math	46.1	31.4	27.7	16.0	18.9
Missing premise	14.7	11.8	10.6	2.1	3.8
Misunderstanding of Mathlib	12.7	15.7	17.0	52.1	51.9
Implicit premise	8.8	11.8	8.5	16.0	11.3
Misunderstanding of Lean	4.9	2.0	0.0	5.3	5.7
Adding extra case	4.4	23.5	25.5	6.4	4.7
Adding extra conclusion	3.4	3.9	10.6	2.1	2.8
NL source ambiguous	4.4	0.0	0.0	0.0	0.0
NL source incorrect	0.5	0.0	0.0	0.0	0.0
Other	0.0	0.0	0.0	0.0	0.9
Provability (% of unfaithful; $n = 204/51/47/94/106$)					
False	51.0	15.7	17.0	27.7	24.5
True	48.5	80.4	80.9	36.2	40.6
Unknown	0.5	3.9	2.1	36.2	33.0
Unprovable	0.0	0.0	0.0	0.0	1.9

Table 7. **Root-cause error profile of autoformalizer outputs vs. the human ProofNet ground truth.** Each autoformalizer column gives the model the informal statement only; the candidate is then classified by the same audit judge used for Table 1, against the ProofNet-Verified gold. The *ProofNet* / *ProofNet-V* columns use the original vs. corrected informal prompt. Cells are percentages (raw counts in the released per-item labels). Faithfulness rises monotonically with model strength—human $44\% <$ qwen3.5-397B $\sim 72\%$ $<$ claude-opus-4.6 $\sim 86\%$—and all models far exceed the human formalizers. The error *shape* is model-dependent: the strong model's unfaithful tail is dominated by *misunderstanding of informal math* (like humans) plus a distinctive *adding extra case* over-specification ($\sim$24–26% vs. 4% human, almost all weaker-but-true), whereas the weaker model is dominated by *misunderstanding of Mathlib* ($\sim 52\%$, $\sim 4\times$ the others) and produces far more *false* or *unjudgeable* statements (false + unknown $\approx$58–64% vs. $\sim 19\%$ for the strong model). These are LLM-judge faithfulness rates, not BEq+ (cf. Table 6).

Benchmark	Items	faithful	stronger	weaker	incomp.	other
PutnamBench	672	640	6	19	7	0
ProverBench	325	130	41	75	79	0
CombiBench	100	72	3	5	14	6
LeanCat	100	83	7	7	3	0
FATE-M	150	135	9	6	0	0
FATE-H	100	95	1	3	1	0
FATE-X	100	86	5	8	1	0

Table 8. Verdict distributions for six non-ProofNet benchmarks (LLM judge only). "Other" includes CombiBench's 6 uncompilable items.