Formal-AVS: A Lean 4 Benchmark for Anytime-Valid Confidence-Sequence Theorem Proving

Aidan Z.H. Yang [1]

Abstract

Anytime-valid confidence sequences provide confidence intervals valid at every stopping time, removing the optional-stopping problem in A/B testing and sequential clinical trials. Deploying them correctly requires proving that the implementation matches the theory, since subtle bugs silently invalidate coverage guarantees. We release Formal-AVS, a benchmark of 60 Lean 4 theorems formalizing properties of four anytime-valid families (Howard-Ramdas, betting, Whitehouse vector, asymptotic CLT) against a companion library and a fixed Mathlib commit. We evaluate seven solvers across three capability levels. Single-shot generalist closure peaks at 22/60 and drops to zero on cross-function targets. A four-round agentic refinement loop closes the first non-Aristotle cross-function target and narrows the prompt-sensitivity gap. Harmonic Aristotle closes 53/60 axiom-clean and surfaces three false-as-stated targets that all other solvers accepted. We release the benchmark, a companion Lean library, and a 14-session Aristotle history archive with closed proofs, refutation witnesses, and axiom audits.

1. Introduction

A/B tests on tech platforms run in the thousands per day. The classical fixed-horizon t-test assumes the sample size is chosen before looking at the data. In practice analysts peek. They stop experiments that look significant or extend those that look close, which can inflate the false-positive rate (Johari et al., 2017). The same *optional stopping* problem arises in clinical trials, where interim analyses are essential but classical group-sequential designs require the number of looks to be fixed in advance. Anytime-valid confidence sequences (Howard et al., 2021; Waudby-Smith & Ramdas, 2024; Whitehouse et al., 2025) solve this: they provide confidence intervals valid at every stopping time. The non-asymptotic flavour (Howard-Ramdas, betting, Whitehouse vector) gives hard guarantees at every t. The asymptotic flavour (time-uniform CLT) gives the large-sample behaviour that governs deployment-scale runtime.

The theory is clean. The code that computes the confidence band is not. A missing absolute value, an off-by-one on a log, or a dropped symmetry hypothesis can each invalidate the coverage guarantee while the software still appears to run. A Lean 4 proof pinned against a fixed Mathlib commit removes that class of error. Writing these proofs by hand is labour-intensive: each theorem in the companion library required multiple iterations of tactic refinement against a changing Mathlib API. Language-model drafters could reduce this effort, but existing Lean-proof benchmarks (MiniF2F (Zheng et al., 2022), ProofNet (Azerbayev et al., 2023), Verina (Ye et al., 2025)) measure closure against frozen statement libraries. They do not test whether a drafter can navigate a library with its own conceptual structure: custom definitions, multi-module import chains, and domain-specific lemma vocabulary. Which drafter capabilities are required to close theorems about such a structure, and can those capabilities be decomposed into prompt-level interventions?

To answer this we introduce Formal-AVS, a benchmark of 60 Lean 4 theorems on anytime-valid confidence sequences across four families. The targets range from single-function sign lemmas to cross-function inequalities that require composing rate definitions from all four families simultaneously. A companion Lean library (`formal-avs-lean`,[1] Apache-2.0, ∼3k lines) supplies the rate functions Mathlib does not yet provide. The benchmark is paired with a 14-session Harmonic Aris-

[1]Athanor AI. Correspondence to: Aidan Z.H. Yang <aidan@athanorl.com>.

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]`https://github.com/athanor-ai/ai4math-2026-kairos-replication`

totle (Achim et al., 2025) history archive containing closed proofs, counterexample witnesses, and axiom audits. Prior Lean benchmarks release target statements or solver traces. Formal-AVS releases both.

We evaluate four closed-source generalists, two open-weights provers (DSPv2-7B (Ren et al., 2025), Goedel-Prover-V2-8B (Goedel-LM Authors, 2025)), and Harmonic Aristotle across three capability levels (single-shot, agentic, unbounded). Single-shot generalist closure peaks at 22/60 (Gemini, hint-list prompt) and drops to zero on cross-function targets. A four-round agentic loop (`cycle_prove` r=4 (Freer & Math Inc., 2025)) closes the first non-Aristotle cross-function target under agentic mode and narrows the prompt-sensitivity gap: Gemini's hint-list advantage vanishes (19 = 19) under agentic mode, suggesting the refinement loop partially substitutes for prompt engineering. Aristotle, under an unbounded compute budget, closes 53/60 axiom-clean and surfaces three false-as-stated targets that all other solvers accepted (§4.4). We also report a prompt-sensitivity matrix and a T3 hint ablation (Table 2).

The benchmark is organized into six tiers. T0–T3 are defined by proof shape before any solver is run. T4 and T5 are empirical failure categories assigned after evaluation. Tier details and a representative target are in §3.3 and Appendix C.

2. Background and related work

Anytime-valid confidence sequences. A confidence sequence $(C_t)_{t\geq 1}$ at level $1-\alpha$ for a parameter μ is a sequence of random intervals satisfying $\mathbb{P}(\forall t \geq 1 : \mu \in C_t) \geq 1-\alpha$. The four canonical constructions for bounded-mean or sub-Gaussian settings each specify an exponential supermartingale $M_t = \exp(\lambda S_t - \psi_F(\lambda)t)$ with a family-specific cumulant-bound function ψ_F. Ville's inequality applied to M_t produces a $1-\alpha$ CS with half-width $\eta_F(\alpha, t) = (\psi_F^{-1} \circ \mathrm{lin})(\alpha, t)$. The benchmark targets theorems about these four η_F (Howard et al., 2021; Waudby-Smith & Ramdas, 2024; Whitehouse et al., 2025), their monotonicity, sign, and cross-family inequalities.

Lean proof benchmarks. MiniF2F (Zheng et al., 2022) covers competition arithmetic. ProofNet (Azerbayev et al., 2023) covers undergraduate analysis, algebra, and topology. Verina (Ye et al., 2025) targets verifiable code generation with 189 Lean tasks. FormalMATH (FormalMATH Authors, 2025) contains 5,560 Lean 4 theorems from olympiad to undergraduate level. LeanDojo (Yang et al., 2023) extracts training and evaluation theorems from Mathlib. REAL-Prover (REAL-Prover Authors, 2024) and LeanAgent (LeanAgent Authors, 2024) mine Mathlib-adjacent undergraduate content. Formal-AVS differs in target domain: a Lean benchmark focused on a research subfield not currently represented in Mathlib, with axiom-audited dependency proofs and a cross-family inequality tier.

Harmonic Aristotle and prover-history archives. The companion archive uses Harmonic's Aristotle prover (Achim et al., 2025), a closed-source proprietary Lean 4 proof-search engine with axiom-audit and counterexample-extraction hooks, and distinguishes genuine closures from drafter hallucinations via machine-checked Lean kernel certificates. Aristotle is not an open-weights model. We accessed it through the public API at `aristotle.harmonic.fun` at no cost during benchmark construction. Closure verdicts and counterexample witnesses are reproducible from the released session IDs, but re-running an Aristotle session against the released library requires a Harmonic API key. The closest prior artefact is LeanDojo's tactic-trace dataset. Our companion artefact differs in three ways: (i) Aristotle sessions rather than Mathlib commits, (ii) refutation traces with counterexample witnesses, and (iii) per-session axiom-audits (Lean `#print axioms` output). The third property is important for evaluating prover trustworthiness: drafter-produced Lean proofs may close the goal but widen the axiom set (Yang et al., 2023).

Hint-ablation methodology. Prompt-hint ablation for Lean drafters has been studied in the context of tactic-premise selection (Polu & Sutskever, 2020; Han et al., 2022; Jiang et al., 2023) and proof-sketch scaffolding. Most prior studies treat hints as a single "context" axis and measure gross closure-rate gain. Formal-AVS decomposes the hint into two orthogonal components, a log-domain substitution hint (*lateral re-encoding*) and a Mathlib-lemma-chain hint (*tactic vocabulary*), and measures their effects independently across four drafter classes plus one specialised 7B prover. The 3-column ablation (§4) shows neither component alone is sufficient. Closure rates increase only when both are supplied to mid-tier generalists.

Agent-style evaluation of code and proof drafters. Single-shot benchmarks understate drafter capability: the same models close many more issues when granted file navigation, multi-step tool calls, and execution feedback, as shown by SWE-bench (Jimenez et al., 2024; Chowdhury et al., 2024). Agentic frameworks for theorem proving and code/database editing are emerging in parallel (Liu et al., 2024; Wang et al., 2024). Formal-AVS supports both completion- and

agent-style evaluation. A target's JSON record exposes the tier shape, required imports, and the $\mathcal{A}_{\text{core}}$ axiom audit, which are the inputs an agentic evaluation loop consumes when delegating to a Lean kernel. The headline sweep is the completion-style baseline against which the agentic-mode gain reported in Table 2 is measured.

Anytime-valid CS formalisations. No prior public Lean 4 library formalises the HR-family, betting family, Whitehouse vector, or asymptotic CLT confidence-sequence families. The mathlib `Probability.Martingale` module (The Mathlib Community, 2026) supplies Doob decomposition and Azuma's inequality but does not instantiate the four CS families or their sharp-constant comparison inequalities. The `formal-avs-lean` library (released with this benchmark) instantiates all four families, supplies Ville's inequality and the quantization-transport lemma, and exposes 29 theorems that require library-specific definitions.

3. Benchmark Construction

We define a *target* as a single Lean 4 theorem statement that a solver is asked to close. Formal-AVS contains 60 targets (§3.3). Each is annotated with its tier, required imports, family, and per-drafter closure record. Appendix C (Listings 1 and 2) shows one target as Lean source and as a JSONL row. The field-level schema is in Appendix E.

3.1. The companion library: `formal-avs-lean`

We construct `formal-avs-lean`,[2] an Apache-2.0 Lean 4 library released as a prerequisite artefact for this benchmark. It instantiates the four canonical anytime-valid confidence-sequence constructions, supplies per-family rate functions η_F, and proves the dependency lemmas required to even *state* the cross-family inequalities used in T3. The asymptotic CLT family is included for comparability with deployed systems that mix non-asymptotic and asymptotic CS constructions. The library is organised into eight modules (`Basic`, `HowardRamdas`, `WaudbySmithBetting`, `WhitehouseVector`, `AsymptoticCLT`, `Quantization`, `PhiTransform`, `MatchingConstants`). Target theorems are extracted from each.

Formal-AVS contains 60 targets by design: target curation is bounded by the ~ 50 axiom-clean dependency lemmas a library author can hand-prove, which allows manual audit of axiom use. Translation pipelines sourcing olympiad or textbook content cannot reach the cross-family tier because Mathlib does not instantiate the rate functions.

Why `formal-avs-lean` was necessary. Mathlib's `Probability.Martingale` supplies Doob decomposition and Azuma's inequality but does not instantiate the four CS families, the per-family rate functions, or Ville's inequality for the betting-process construction. Without those, neither generalist drafters nor Aristotle can attempt the T3 cross-function statements: the goal type $\eta_{\text{HR}}(b) \leq \eta_{\text{vector}}(b)$ is not expressible. The library closes that gap in $\sim$3k lines across 8 modules and ~ 50 dependency lemmas, with the QA guarantees described in §3.2.

3.2. Quality assurance and axiom audit

Because the benchmark grades drafters against the library, the library itself must be trustworthy: a hidden `sorry` or a silently-widened axiom set in any dependency lemma would invalidate every closure built on top. Quality assurance has four components. **(i) Axiom-clean baseline.** Every reference proof, every closed drafter attempt, and every closed Aristotle session terminates inside the canonical Mathlib core $\mathcal{A}_{\text{core}} = \{\texttt{propext}, \texttt{Classical.choice}, \texttt{Quot.sound}\}$. The drafter-sweep closure criterion (§3.3) rejects any candidate whose `#print axioms` dump extends beyond $\mathcal{A}_{\text{core}}$, and per-target dumps union to exactly $\mathcal{A}_{\text{core}}$ (Appendix K.5). This rules out the `axiom my_lemma : ...` shortcut. **(ii) Hand-authored dependency lemmas.** The ~ 50 dependency lemmas (Ville's inequality, quantization-transport, Φ-substitution, sharp constants) were written by the library authors before any solver run, and each was manually cross-checked against the published constructions of Howard et al. (Howard et al., 2021), Waudby-Smith and Ramdas (Waudby-Smith & Ramdas, 2024), and Whitehouse et al. (Whitehouse et al., 2025). **(iii) Refutation discovery.** Three benchmark candidates authored as legitimate targets were refuted by Aristotle on the first attempt, and both original and corrected forms ship in the JSONL (§4.4). **(iv) Toolchain pinning** (§I): library, JSONL, and verification driver are pinned to a fixed Lean toolchain and Mathlib commit, and `bash reproduce.sh` re-runs the full audit on cached outputs.

The dependency lemmas are sorry-free. Target theorems contain `sorry` by design: these are the statements solvers attempt to close, and the 53/60 closed proofs are in the Aristotle history and drafter output files.

[2] Source at `https://github.com/athanor-ai/ai4math-2026-kairos-replication`.

3.3. Tiers and failure categories

Each target carries two kinds of label. The four *tiers* T0-T3 are assigned by theorem shape before any drafter run. The two *empirical failure categories* T4 and T5 are assigned after evaluation, because their definitions reference observed solver behaviour. We separate the two kinds of label so that the tier signal is independent of solver outcome.

Pre-evaluation tiers (T0-T3). **T0** (Mathlib canon, 7 theorems): statement and proof expressible entirely within Mathlib (sanity-check floor). **T1** (single-function friendly, 34): sign / monotonicity / positivity of a single η_F closeable by `positivity`, `nlinarith`, or a short `simp`-chain. **T2** (single-function challenging, 8): single-function targets whose closure requires a substitution or a Mathlib lemma not named in the goal. **T3** (cross-function inequality, 6): inequalities or sharp-constant identities between two rate functions (e.g. $\eta_{\text{vector}} = \sqrt{2}\,\eta_{\text{HR}}$), admitting a log-domain Φ-substitution. T0-T3 assignment depends only on theorem signature and is fixed before any drafter sweep.

Empirical failure categories (T4, T5). **T4** (library-gap, 4 theorems): targets where no evaluated solver returned a closure and the missing piece is a Mathlib API not yet upstream (sub-Gaussian MGF bound for martingale sums, Ville's inequality for supermartingales, vector concentration, time-uniform CLT). A Mathlib pull request adding the required API would move these targets out of T4. **T5** (unresolved, 1 theorem): the sole T5 target, `equivalence_break_at_finite_precision`, has zero close-rate across every evaluated solver. It is a witness-construction problem on the continuous-equivalence break between Howard-Ramdas and Ville thresholds at finite precision. No missing Mathlib API explains the failure. T4 and T5 are empirical categories from this evaluation, not predictions for future solvers.

3.4. Drafter sweep protocol

Each drafter is evaluated at $N = 5$ attempts per target under a mixed-temperature schedule: one greedy attempt (temperature $T = 0$) followed by four independent samples at $T = 0.7$. The attempt budget is 8k tokens per sample. We define pass@N at the target level. A target is *closed* for a drafter if at least one of the N attempts produces a Lean 4 proof that (a) type-checks on the specified toolchain, (b) closes the goal with no residual `sorry`, and (c) does not widen the axiom set beyond {`propext`, `Classical.choice`, `Quot.sound`}. Pass@N is defined at the target level: a target is closed if any of the N attempts succeeds. We do not aggregate over individual attempts as if they were independent samples.

Target-selection protocol. We extracted target theorems from the eight `formal-avs-lean` modules. The selection criteria, applied in order, were the following. (i) The theorem is provable in Lean 4 against Mathlib at the pinned commit, modulo the dependency lemmas already proved inside `formal-avs-lean`. (ii) The statement does not import a non-Apache-2.0 third-party library. (iii) The statement is at most 400 characters and does not require a custom tactic block to state. We did not pre-screen for drafter difficulty. Of the ~ 120 candidate theorems in the library, 60 met the criteria and form the released benchmark. The remaining candidates were excluded for reasons recorded per-target in the supplementary `selection_log.csv`.

Evaluated slate. The headline drafter sweeps were run on all 60 targets. The 60-target benchmark is the canonical evaluation set in this submission. The benchmark includes 4 T4 entries, 1 T5 entry, and 7 supplementary T0 entries (see Table 1). T4 and T5 entries are zero by construction in the 60-target table because they were assigned to those categories on the basis of zero closures across evaluated solvers (§3.3). We report headline rates on all 60 targets.

Drafter set. Four closed-source drafters are scored in the headline sweep. Claude Sonnet 4.6, Kimi K2.5, and Gemini 3 Pro under a shared hint-list prompt. Mistral Large 3 under a neutral prompt (§4.2 reports the prompt-sensitivity result). Mistral collapses to 0/300 on the same hint-list prompt because of greedy hint-list compliance, so the headline row uses the prompt that produces a non-degenerate row. Cross-drafter aggregates over Mistral and the other three should be read with this prompt asymmetry in mind. A supplementary closed-source row, Claude Opus 4.6, is reported on all 60 targets at $N = 5$ under the hint-list prompt. Two specialised open-weights drafters complete the headline sweep: DeepSeek-Prover-V2-7B (Ren et al., 2025) (huggingface.co/deepseek-ai/DeepSeek-Prover-V2-7B) and Goedel-Prover-V2-8B (Goedel-LM Authors, 2025) (huggingface.co/Goedel-LM/Goedel-Prover-V2-8B). Both are scored on all 60 targets at $N = 5$ under the same neutral specialised-prover prompt template. Community quantization choices and reproduction details are in Appendix G. DSPv2 attempts are released as cached candidates in the supplementary archive. The Goedel-V2 row is reproducible end-to-end from the released weights and evaluation driver.

3.5. Hint ablation protocol

Three prompt columns per T3 target are evaluated, with names chosen so the content is recoverable from the label. **Base**: goal statement only (no-hint baseline). **Sub**: Base plus a one-line Φ-substitution hint $x \mapsto \exp(\lambda^\star x - (\lambda^\star)^2\sigma^2/2)$, with family-specific $\lambda^\star$ (isolates lemma search given the substitution). **Sub+Chain**: Sub plus an explicit Mathlib lemma chain (`Real.exp_le_one_iff`, `Real.log_nonneg`, `Real.sqrt_le_sqrt`, ...) with usage-direction annotations (isolates lemma application given both). A boxed-table form is Table 4 (Appendix B). The five-stage evaluation pipeline is expanded as Table 5 in Appendix D.

The Sub+Chain lemma list was reverse-engineered from an Aristotle T3 closure (§3.3, footnote on hint-chain provenance). An alternative library-reference chain (§G.3) is released as a sensitivity check, overlapping on 5/7 lemmas. We read Sub+Chain as a proof-plan-condition upper bound on drafter T3 closure given the lemma chain, not as an independent tactic-vocabulary measurement.

3.6. Aristotle history collection

Fourteen Harmonic Aristotle (Achim et al., 2025) sessions were collected between 2026-03 and 2026-04 against the same `formal-avs-lean` library. Each session submits 1 to 5 targets. We record per-target verdict (`closed`, `partial`, or `refuted`), the closed Lean proof when applicable, the counterexample when applicable, the axiom-set dump of the closed Lean kernel, and the fingerprint hash of the submitted Lean tarball. All 14 sessions are released as `aristotle_history.jsonl` alongside the benchmark. Reproducing a session against the released library requires a Harmonic API key (we accessed Aristotle through the public dashboard at `aristotle.harmonic.fun`).

Refutation handling. Three benchmark targets were authored as legitimate library statements but Aristotle returned counterexamples on first attempt, identifying that the as-stated forms were mathematically false (§4.4). We retain all three in the benchmark JSONL. The released schema includes both the corrected theorem (used in headline drafter sweeps and the Aristotle 53/60 closure count) and the original refuted form (used to support the statement-correctness classification axis). Refutations were not seeded by the authors. They are a discovered property of the library that the benchmark surfaces and preserves.

Aristotle compute budget. Each Aristotle session is a single API call submitted via `aristotle submit` against the released library. We did not impose a per-session token or rollout cap. Sessions return on average within 30 to 50 minutes of wall time. Drafters are scored at $N = 5$ attempts of 8k tokens each. The Aristotle vs. drafter comparison is therefore between a proof-search system with unbounded internal compute and a generalist drafter at a fixed token budget. The 53/60 vs. 2/24 T3 gap conflates training data, training scale, search compute, and prompt-conditioned tactic vocabulary, and is reported as such.

Reproducibility. The supplementary archive includes the `formal-avs-lean` library at the cited Lean toolchain and Mathlib pin (toolchain reference and Mathlib commit are listed in the supplementary `REPRODUCE.md`), the benchmark JSONL with both corrected and original (refuted) statement forms, the prompt files, the drafter sweep driver, the Wilson-CI computation script, and the Lean kernel verification script that recomputes the {`propext`, `Classical.choice`, `Quot.sound`} axiom audit. A single command (`bash reproduce.sh`) runs the verification pipeline on cached drafter outputs to regenerate every numeric cell in the headline tables. Drafter output files for all evaluated cells are released in the archive.

4. Results

4.1. Per-tier results on the 60-target benchmark

Table 1 reports per-drafter closure rates at $N = 5$ on the 60-target benchmark, plus a 60-target status table. T4 and T5 entries are zero in the 60-target table by construction (§3.3). Aristotle's closure rate is in §4.1. Most drafter closures fall on T1. The other five tiers show low closure.

T0 (0/28 across 4 generalists): generalists fail on Mathlib-canon lemmas despite these appearing in training corpora. The failure mode is a tactic-prior mismatch: drafters attempt multi-step rewrites where a single `simp` or `exact?` suffices, because the `formal-avs-lean` namespace context shifts the tactic distribution away from the Mathlib patterns the models memorised (§K.11). **T2** (0/8): requires a substitution or unnamed Mathlib lemma. **T3** (2/24): drafter close-rate 0–1/6. Aristotle closes 6/6. The gap is a system difference, not per-token (§4.3). **T4** (0/4): blocked by missing Mathlib APIs (Ville, sub-Gaussian MGF, vector concentration, time-uniform CLT). **T5** (0/1): Aristotle refuted the as-stated form (§4.4). Corrected

form unresolved.

Capability ladder: single-shot → agentic → Aristotle. Three capability levels are evaluated on the same slate.

Hint ablation (T3). A 3-column T3 hint study (Base, Sub, Sub+Chain) is reported in Appendix A. Generalist closure on T3 rises from 0–1/6 at Base to 3–5/6 at Sub+Chain. The Sub+Chain chain was reverse-engineered from an Aristotle closure, so we read it as an upper bound (sensitivity check in Appendix G.3).

4.2. Prompt-sensitivity is drafter-specific and bidirectional

Table 2 reports per-drafter closure rate under both the shared hint-list prompt (which enumerates nine allowed Mathlib tactics as suggestions and asks for minimal proofs) and a neutral prompt that omits the enumeration ("Close this theorem with a clean Lean 4 proof. Use any Mathlib tactic."). Both evaluated on the same 60-target slate at $N = 5$ (one greedy $T = 0$ + four $T = 0.7$).

The hint-list helps three of four generalists and reduces Mistral (2/60 → 0/60). Sonnet's sensitivity is largest (6×). Extended per-drafter analysis in Appendix L.[3]

4.3. Capability ladder: single-shot, agentic, and Aristotle

Closure rate increases monotonically across three capability levels on the same slate: *single-shot* (best-of-N, no error feedback), *agentic* (draft → `lake build` → feed errors back → refine, r=4 rounds), and *Aristotle* (unbounded internal refinement). Harmonic Aristotle (Achim et al., 2025) closes 53/60 axiom-clean (7/7 T0, 33/34 T1, 7/8 T2, 6/6 T3) plus 3 refuted (§4.4). Companion-archive closures are in §K.2. Under `cycle_prove` r=4:

- **Gemini 3 Pro.** Single-shot neutral 13/60 → agentic 19/60 (46% lift). Gemini agentic closes `ranking_four_way` (T3) on round 2: a non-Aristotle agentic closure on a T3 target.
- **Sonnet 4.6.** Neutral 3/60 → agentic 15/60 (5× lift from the refinement loop alone).
- **Goedel-V2-Q6_K.** Single-shot 5/60 → agentic 13/60 (2.6× lift). Goedel-V2 was trained with verifier-guided self-correction (Goedel-LM Authors, 2025): the model uses Lean compiler feedback to iteratively revise its proofs. Under `cycle_prove` this training activates and Goedel surpasses DSPv2's single-shot 9/60, becoming the strongest local prover on our slate at $0 marginal cost.

The prompt-sensitivity gap narrows under cycle_prove. Sonnet's single-shot 6× hint-list advantage (3/60 → 18/60) reduces to 15% under agentic (13 → 15). Gemini's advantage disappears (19 = 19). Per-target lake errors provide stronger signal than a static tactic list, partially substituting for prompt engineering.

Across both drafters, 14 targets close under both, 5 are Gemini-only (including 2 T3 cross-function targets), 1 is Sonnet-only. Most closures land on rounds 1–2 (28/34). Rounds 3–4 contribute 6/34: the refinement loop's value is primarily one error-correction cycle, not deep multi-round search. Generalist single-shot peaks at 22/60, agentic r=4 reaches 19/60, and Aristotle's open-ended budget closes 53/60. The hint ablation (Table 3) shows the tactic vocabulary is latent in the generalist drafters. The gap is attributable to search depth, not missing knowledge.

4.4. Refutation findings

Beyond closure, Aristotle refuted three benchmark targets whose as-stated form is mathematically false, returning machine-checkable counterexamples on each: `equivalence_break_at_finite_precision` (T5, missing symmetry hypothesis), `slack_antitone_in_scale` (T2, missing bits-match hypothesis, counterexample `bp`$_1$`=(1,1)`, `bp`$_2$`=(2,1)`), and `div_le_div_of_nonneg_left` (T1, typo in direction: `a=1,b=2,c=3` gives `1/2 > 1/3`). Each corrected form was subsequently proved.

Provenance of the three refutations. These statements were authored as legitimate library targets prior to any solver run. Aristotle subsequently flagged each as-stated form as false on its first attempt. The refutations were not seeded into the benchmark by the authors. The benchmark JSONL contains the corrected forms in the headline target column and the original false-as-stated forms in a parallel `refuted_form` field.

The mix of correct, borderline, and false-as-stated targets therefore supports a second evaluation axis: *statement-correctness classification* via refutation. Pass@N solvers that do not attempt counterexample construction miss this axis entirely.

[3] Four extended analyses are in Appendix K: Aristotle history, companion closures, cross-drafter tactic diversity, and the no-closure tail.

Table 1. Per-drafter closure rates on the 60-target benchmark at N=5 (one greedy + four T=0.7 samples). Overall column sums the 60-target benchmark (T4/T5 zero by construction). Wilson CIs in Appendix 7. See Appendix I.1 for N=10 results.

Drafter	T0	T1	T2	T3	T4	T5	overall pass@1	pass@5	pass@10
Claude Sonnet 4.6	0/7	18/34	0/8	0/6	0/4	0/1	11/60	18/60	18/60
Kimi K2.5	0/7	17/34	0/8	1/6	0/4	0/1	13/60	18/60	18/60
Gemini 3 Pro	0/7	22/34	0/8	0/6	0/4	0/1	9/60	22/60	23/60
Mistral Large 3	0/7	1/34	0/8	1/6	0/4	0/1	0/60	2/60	2/60
Claude Opus 4.6	0/7	13/34	0/8	0/6	0/4	0/1	13/60	13/60	14/60
DSPv2-7B GPTQ-int8	3/7	8/34	0/8	1/6	0/4	0/1	10/60	12/60	13/60
Goedel-Prover V2-Q6_K	2/7	3/34	0/8	0/6	0/4	0/1	4/60	5/60	6/60

Table 2. Capability ladder: prompt-sensitivity across single-shot, agentic, and Aristotle on the 60-target benchmark (pass@5). Single-shot uses N=5 (1 greedy + 4 stochastic). Agentic uses `cycle_prove` r=4 (draft → `lake build` → refine, default from OpenGauss (Freer & Math Inc., 2025)). Hint-list enumerates nine Mathlib tactics. Neutral omits the enumeration. Aristotle (Achim et al., 2025) operates under an unbounded compute budget (no prompt distinction). The hint-list advantage shrinks under agentic mode: Sonnet's 6× single-shot lift (3 → 18) reduces to 15% (13 → 15). Gemini's advantage vanishes (19 = 19): per-target lake errors substitute for the static tactic enumeration.

Drafter	Hint-list	Neutral
Single-shot (N=5)		
Gemini 3 Pro	22/60	13/60
Claude Sonnet 4.6	18/60	3/60
Kimi K2.5	18/60	13/60
DSPv2-7B	12/60	9/60
Goedel-V2-Q6_K	5/60	3/60
Claude Opus 4.6	13/60	4/60
Mistral Large 3	0/60	2/60
Agentic (`cycle_prove` r=4)		
Gemini 3 Pro	19/60	19/60
Claude Sonnet 4.6	15/60	13/60
Goedel-V2-Q6_K	10/60	13/60
Aristotle (unbounded)	53/60	

5. Discussion

The generalist–Aristotle gap as a capability measurement. The gap between the best generalist (22/60, Gemini single-shot) and Aristotle (53/60) quantifies the capability gap this benchmark is designed to measure. Agentic mode narrows the prompt-sensitivity gap but does not close the closure gap: Gemini agentic reaches 19/60, still far below 53/60. The remaining distance is not a prompt problem. Cross-function targets (T3) require composing definitions from multiple library modules in a single proof step. Generalists trained on Mathlib do not learn this composition because Mathlib contains no cross-family CS inequalities. The T3 hint ablation (Appendix A) confirms that supplying both the substitution and the lemma chain closes the gap (3–5/6 at Sub+Chain), but the drafter cannot produce those hints on its own.

Current provers are memorisation engines, not reasoners. Goedel-Prover-V2 (Goedel-LM Authors, 2025), trained on Mathlib, responds to `etabetting_nonneg` with *"the Lean 4 code provided does not define etaBetting . . . we cannot proceed with a meaningful proof."* This is not a failure of tactic knowledge. It is a failure to read a new definition and reason about it. Any domain-specific Lean library outside Mathlib will surface this gap: the prover has seen the tactics but never the definitions. Retrieval-augmented proving, fine-tuning on domain libraries, or longer-context in-context learning are candidate next steps. The benchmark provides the measurement tool. The T1 tier (34 single-function targets against custom definitions) is designed to isolate this axis.

Refutation as a safety signal. Aristotle surfaces three false-as-stated targets that every other solver accepted. Two are definitional errors (an off-by-one in a stopping-rule admissibility condition and a missing monotonicity hypothesis); one is a quantitative counterexample. In all three cases, the generalist drafters produce confident, syntactically valid Lean proofs that the kernel rejects only because the statement is false. A deployment pipeline that trusts the drafter and skips kernel verification would ship a broken guarantee. The refutation track measures a capability axis absent from pass@N benchmarks: whether a solver can identify that a statement is wrong, not just fail to prove it.

Limitations. The benchmark covers one subfield (anytime-valid CS) with 60 targets. Closure rates may not transfer to other Lean libraries with different import structures or proof shapes. Aristotle is a closed-source system. Its results are reproducible via session IDs but not by re-running an open model. The 60-target scale limits statistical power on per-tier comparisons. Wilson

95% CIs and contamination notes are in Appendix 7.

Broader impact. Anytime-valid confidence sequences are deployed in production A/B testing and clinical trial monitoring systems that make decisions affecting millions of users. A silent coverage-guarantee violation in these systems can lead to incorrect treatment decisions or product launches based on false statistical significance. Formally verified implementations remove this failure class. This benchmark measures how close current language-model provers are to automating that verification, and the refutation track provides an early warning for the complementary risk: solvers that produce plausible-looking proofs of false statements.

References

Achim, T., Best, A., Bietti, A., Der, K., Fédérico, M., Gukov, S., Halpern-Leistner, D., Henningsgard, K., Kudryashov, Y., Meiburg, A., Michelsen, M., Patterson, R., Rodriguez, E., Scharff, L., Shanker, V., Sicca, V., Sowrirajan, H., Swope, A., Tamas, M., Tenev, V., Thomm, J., Williams, H., and Wu, L. Aristotle: IMO-level automated theorem proving, 2025. URL `https://arxiv.org/abs/2510.01346`.

Azerbayev, Z., Piotrowski, B., Schoelkopf, H., Ayers, E. W., Radev, D., and Avigad, J. ProofNet: autoformalizing and formally proving undergraduate-level mathematics. In *arXiv preprint*, 2023.

Chowdhury, N., Aleman, J., Wu, J., and Edwards, C. Introducing SWE-bench Verified: a human-validated subset of SWE-bench. OpenAI technical report, 2024.

FormalMATH Authors. FormalMATH: benchmarking formal mathematical reasoning of large language models, 2025.

Freer, C. and Math Inc. OpenGauss: multi-agent Lean 4 theorem proving via cycle-engine refinement. `https://github.com/math-inc/OpenGauss`, 2025. Lean4-skills cycle-engine default: `max_rounds=4`.

Goedel-LM Authors. Goedel-Prover-V2: an open-source language model for Lean 4 theorem proving. Hugging Face model card, `https://huggingface.co/Goedel-LM/Goedel-Prover-V2-8B`, 2025. Princeton-led release. 8B base, GGUF Q6_K quantization used in this benchmark.

Han, J. M., Rute, J., Wu, Y., Ayers, E. W., and Polu, S. Proof artifact co-training for theorem proving with language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Howard, S. R., Ramdas, A., McAuliffe, J., and Sekhon, J. Time-uniform, nonparametric, nonasymptotic confidence sequences. *Annals of Statistics*, 49(2): 1055–1080, 2021. doi: 10.1214/20-AOS1991.

Jiang, A. Q., Welleck, S., Zhou, J. P., Li, W., Liu, J., Jamnik, M., Lacroix, T., Wu, Y., and Lample, G. Draft, sketch, and prove: guiding formal theorem provers with informal proofs. In *International Conference on Learning Representations (ICLR)*, 2023.

Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. SWE-bench: can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024.

Johari, R., Koomen, P., Pekelis, L., and Walsh, D. Peeking at A/B tests: Why it matters, and what to do about it. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1517–1525. ACM, 2017.

LeanAgent Authors. LeanAgent: lifelong learning for formal theorem proving. arXiv preprint, 2024.

Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., Zhang, S., Deng, X., Zeng, A., Du, Z., Zhang, C., Shen, S., Zhang, T., Su, Y., Sun, H., Huang, M., Dong, Y., and Tang, J. AgentBench: evaluating LLMs as agents, 2024.

Polu, S. and Sutskever, I. Generative language modeling for automated theorem proving. In *arXiv preprint*, 2020.

REAL-Prover Authors. REAL-Prover: retrieval-augmented Lean theorem-proving. arXiv preprint, 2024.

Ren, Z. Z., Shao, Z., Song, J., Xin, H., Wang, H., Zhao, W., Zhang, L., Fu, Z., Zhu, Q., Yang, D., et al. DeepSeek-Prover-V2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint*, 2025.

The Mathlib Community. Mathlib4 `Probability.Martingale` module. `https://leanprover-community.github.io/mathlib4_docs/`, 2026.

Wang, K., Ren, H., Zhou, A., Lu, Z., Luo, S., Shi, W., Zhang, R., Song, L., Zhan, M., and Li, H. MathCoder: seamless code integration in LLMs for enhanced mathematical reasoning, 2024.

Waudby-Smith, I. and Ramdas, A. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B*, 86(1):1–27, 2024.

Whitehouse, J., Ramdas, A., Wu, Z. S., and Sutton, R. Time-uniform self-normalized concentration for vector-valued processes. In *Proceedings of the 38th Conference on Learning Theory (COLT)*, volume 291, pp. 5714–5715, 2025.

Yang, K., Swope, A., Gu, A., Chalamala, R., Song, P., Yu, S., Godil, S., Prenger, R., and Anandkumar, A. LeanDojo: theorem proving with retrieval-augmented language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Ye, Z., Yan, Z., He, J., Kasriel, T., Yang, K., and Song, D. Verina: benchmarking verifiable code generation, 2025.

Zheng, K., Han, J. M., and Polu, S. miniF2F: a cross-system benchmark for formal Olympiad-level mathematics. In *International Conference on Learning Representations (ICLR)*, 2022.

A. T3 hint-ablation results

B. Hint-ablation legend table

C. Benchmark format

The two listings below show one T3 target as it appears in the Lean source (Listing 1) and as a row of `benchmark.jsonl` (Listing 2). The full field-level schema is in Appendix E.

Listing 1. Lean theorem statement for the T3 target `etavector_eq_sqrt_two_mul_etahr`. Stored verbatim in the `theorem_statement` field of the matching JSONL row.

```
theorem etavector_eq_sqrt_two_mul_etahr
    (b : Real) (hb : 1 <= b) :
    etaVector b = Real.sqrt 2 * etaHR b := by
  unfold etaVector etaHR
  rw [Real.sqrt_mul]
  . ring
  . positivity
```

Listing 2. `benchmark.jsonl` record (excerpt) for the same target. The `hint_ablation` block carries the three prompt columns for T3 targets only (Table 4).

```
{
  "target_id":
      "etavector_eq_sqrt_two_mul_etahr",
  "tier": "T3",
  "family": "cross",
  "requires_formal_avs_lean": true,
  "required_imports": [
    "FormalAvs.HowardRamdas",
    "FormalAvs.WhitehouseVector"
  ],
  "closure_rate_n5": {
    "sonnet_4_6": 0.0,
    "kimi_k2_5": 0.6,
    "gemini_3pro": 0.0,
    "mistral_l3": 0.4
  },
  "hint_ablation": {
    "BASE": {"sonnet_4_6": 0.0, ...},
    "SUB": {"sonnet_4_6": 0.2, ...},
    "SUB+CHAIN": {"sonnet_4_6": 0.8, ...}
  },
  "aristotle_session_id": "74fe5c49"
}
```

D. Evaluation-protocol pipeline

Table 5 expands the five-stage drafter sweep pipeline summarised in §3.5. Every (target, drafter, prompt) cell traverses the five stages in order. Stage 4 pins outcomes to the fixed logical foundation of §3.2.

E. Dataset schema

Each row of `benchmark.jsonl` contains the following fields.

- `target_id`: unique identifier (stable across versions).
- `theorem_statement`: the Lean 4 theorem statement as a raw string (imports included).
- `tier`: one of `T0`, `T1`, `T2`, `T3`, `T4`, `T5`.
- `requires_formal_avs_lean`: boolean flag. True if the statement references a `formal-avs-lean` definition not in Mathlib.
- `family`: one of `hr`, `vector`, `betting`, `aCS`, `cross` (cross-family inequality).
- `closure_rate_n5`: map from drafter-name to fraction closed at $N = 5$.
- `hint_ablation`: present for T3 targets. Map from hint-column key (`"BASE"`, `"SUB"`, or `"SUB+CHAIN"` as in Table 4) to closure-rate map across drafters.
- `aristotle_session_id`: fingerprint hash linking to a row in `aristotle_history.jsonl` (if present).
- `mathlib_lemmas`: for T2/T3, the Mathlib lemma chain that admits the closure (used in the Sub+Chain prompt).

Table 3. 3-column hint study on T3 cross-function inequalities (pass@5). The Sub+Chain chain was reverse-engineered from an Aristotle T3 closure, so this column is a proof-plan-condition upper bound, not an independent capability test. Generalist closure rises from 0–1/6 at Base to 3–5/6 at Sub+Chain. DSPv2 closes the same single target across both hint columns. Footnote superscripts match Table 1.

Drafter	Base	Sub	Sub+Chain
Claude Sonnet 4.6	0/6	1/6	4/6
Kimi K2.5	1/6	1/6	3/6
Gemini 3 Pro	0/6	2/6	4/6
Mistral Large 3*	0/6	0/6	0/6
Claude Opus 4.6‡‡	0/6	2/6	5/6
DSPv2-7B GPTQ-int8	0/6	1/6	1/6

Table 4. Hint-ablation legend, boxed-table form. Same content as the inline legend in §3.5.

Base	Goal statement only, no hint (no-hint baseline).
Sub	Base plus Φ-substitution hint $x \mapsto \exp(\lambda^\star x - (\lambda^\star)^2\sigma^2/2)$, family-specific $\lambda^\star$ (lemma search given the substitution).
Sub+Chain	Sub plus an explicit Mathlib lemma chain (`Real.exp_le_one_iff`, `Real.log_nonneg`, `Real.sqrt_le_sqrt`, ...) (lemma application given both).

Table 5. Five-stage evaluation pipeline applied per (target, drafter, prompt) cell. Companion to §3.5. Expanded form of the protocol text in §3.3.

#	Stage	Action
1	Target	Lean theorem read from `benchmark.jsonl` (60 targets), tagged with tier T0–T5 and required imports.
2	Prompt	Hint-list or neutral template. For T3, one of Base / Sub / Sub+Chain (Table 4).
3	Sample	N=5 attempts per cell: 1 greedy at T=0 plus 4 samples at T=0.7, 8k tokens each.
4	Verify	`lake build` on the candidate, reject any `sorry`, then assert `#print axioms` $\subseteq \mathcal{A}_{\text{core}}$.
5	Score	Target closed for the cell iff ≥ 1 of N attempts passes stage 4. Aggregate as per-cell pass@N.

F. Aristotle history schema

Each row of `aristotle_history.jsonl` contains the following fields.

- `session_id`: short fingerprint (8-12 hex chars).
- `target_id`: pointer into `benchmark.jsonl`.
- `submitted_at`, `completed_at`: ISO-8601 UTC timestamps.
- `verdict`: one of `closed`, `partial`, `refuted`.
- `axiom_audit`: sorted list of axioms used by the closed Lean kernel, as reported by `#print axioms`.
- `counterexample`: present only when `verdict = refuted`. The Lean-syntax counterexample witness.
- `residual_sorries`: list of `sorry` names present in the partial proof (present only when `verdict = partial`).
- `compute_minutes`: elapsed wall-clock time of the Aristotle session.

G. Drafter prompts

G.1. Generalist drafter prompt (closed-source)

```
You are a Lean 4 expert proficient with Mathlib.
Close the following theorem with a clean tactic
    proof.
Prefer concise tactic chains. Avoid ‘sorry‘.
    Avoid axioms.

<theorem-statement>

<optional hint block>
```

The Sub hint adds: “Hint: the log-domain substitution $x \mapsto \exp(\lambda^\star x - (\lambda^\star)^2\sigma^2/2)$ with family-specific $\lambda^\star$ is likely useful.”

The Sub+Chain hint adds on top: “Hint: the following Mathlib lemmas are likely relevant, in approximate

order: `Real.exp_le_one_iff`, `Real.log_nonneg`, `Real.sqrt_le_sqrt`, `Real.exp_log_eq_iff`, `Real.sqrt_mul_self`, `field_simp`, `positivity`."

G.2. Specialised prover prompts (DSPv2, Goedel-V2)

DSPv2 is autocomplete-trained: the input terminates at the trailing `:=` token and the model emits the proof body directly.

```
<imports>

<opens>

namespace <scratch_namespace>

<theorem-header> :=
```

Goedel-V2 is chat-trained: per the model card, the input is wrapped as a single user turn that asks for a proof plan followed by the final code in a fenced `lean4` block. We extract the proof body from that block and verify against the same lake project as the rest of the sweep.

```
Complete the following Lean 4 code:

‘‘‘lean4
<imports>

namespace <scratch_namespace>

<theorem-header> := by
  sorry
‘‘‘

Before producing the Lean 4 code to formally
    prove the given
theorem, provide a detailed proof plan ...
```

The Sub+Chain column adds the Mathlib-lemma hint as a Lean `/- hint -/` block before the trailing `:=` token (DSPv2) or in the user turn before the code block (Goedel-V2). DSPv2 is prefix-only and ignores non-prefix hint context. We retain all three hint columns for both rows for parity with the closed-source generalists.

We use community quantizations: DSPv2 GPTQ-int8 (one of the 15 community variants linked from the official model page), Goedel-V2 GGUF Q6_K. Both are inference-only artifacts. The underlying weights are the official 7B / 8B releases.

G.3. Alternative T3 hint chain from library reference proofs

To rule out the concern that the Sub+Chain hint contaminates the ablation by replaying Aristotle's tactic-prior, we extracted a second hint chain directly from the closed reference proofs in `formal-avs-lean` (the per-target proofs the library authors hand-wrote for the T3 statements, prior to any solver run). The two chains are reported below:

Sub+Chain headline. `Real.exp_le_one_iff`, `Real.log_nonneg`, `Real.sqrt_le_sqrt`, `Real.exp_log_eq_iff`, `Real.sqrt_mul_self`, `field_simp`, `positivity`.

Library reference proofs. `Real.sqrt_le_sqrt`, `Real.sqrt_mul`, `Real.log_nonneg`, `Real.le_sqrt_of_sq_le`, `nlinarith`, `field_simp`, `Real.log_two_gt_d9`.

The two chains overlap on 5/7 lemmas (`Real.sqrt_le_sqrt`, `Real.log_nonneg`, `field_simp`, `positivity` or `Real.le_sqrt_of_sq_le`, and a sqrt-multiplication identity). The non-overlapping parts reflect different proof shapes (the Sub+Chain chain includes `Real.exp_log_eq_iff` for the substitution step. The reference-proof chain uses `nlinarith` + `Real.log_two_gt_d9` for numerical bounds). Both chains live in the same Mathlib lemma neighbourhood. The non-overlap reflects different proof shapes, not distribution shift.

The headline ablation in Table 3 uses the Sub+Chain chain. The library-reference chain is released in the supplementary archive as an alternative hint set for researchers wanting an Aristotle-independent baseline.

Spot-check N=5 result. We ran a small Sonnet $N = 5$ comparison on 3 of the 6 T3 targets (`etabetting_le_etahr`, `etavector_eq_sqrt_two_mul_etahr`, `etahr_le_etavector`) using the library-reference chain. Sonnet closed 0/15 attempts, versus the headline Sub+Chain result of 4/6 across all six T3 targets. The result is informative: the specific lemma sequence in the Sub+Chain chain is consequential within the broad "Mathlib lemma chain" hint category. The two chains overlap on 5/7 lemmas but produce different per-target close-rates at small N, suggesting that hint composition (lemma order, presence of specific bridging identities like `Real.exp_log_eq_iff`) is itself a capability axis. A full $4 \times 6 \times N = 5$ comparison across all four generalist drafters and all six T3 targets is deferred to a future revision.

H. Tier-distribution table

I. Baseline reproducibility

To reproduce the headline sweep:

Table 6. Target distribution across tier × family. 60 targets total. T4 plus T5 are the empirical failure categories (5 theorems). 53 of 60 non-frontier targets require `formal-avs-lean`.

Family	T0	T1	T2	T3	T4	T5
HR-family	2	11	3	-	-	-
betting family	1	10	1	-	-	-
Whitehouse vector	1	6	2	-	-	-
Asymptotic CLT	1	7	2	-	-	-
Cross-family (inequality)	-	-	-	6	-	-
Mathlib-only reference	2	-	-	-	-	-
Library-gap (Mathlib PR)	-	-	-	-	4	-
Capability-ceiling frontier	-	-	-	-	-	1
Total	7	34	8	6	4	1

1. Clone the companion Lean 4 library at the pinned release commit (snapshot URL in supplementary `REPRODUCE.md`, item LIB-SNAPSHOT. Lean toolchain and Mathlib commit pin in `lean-toolchain` and `lakefile.lean`).
2. Clone the dataset repo from the snapshot Hugging Face location (URL in supplementary `REPRODUCE.md`, item DATASET-SNAPSHOT).
3. Run the provided `sweep.py` against a drafter endpoint. It submits each of the 60 targets under the mixed-temperature pass@5 schedule (one greedy sample at $T = 0$ followed by four independent samples at $T = 0.7$) and writes per-target closure records. The driver prints the SHA-256 of the prompt files and dataset snapshot it used.
4. Verify each closure by running `lake build` against the produced Lean file. The per-target axiom audit is extracted via `lean --run axiom_audit.lean`.
5. Run `bash reproduce.sh` from the supplementary archive to regenerate every numeric cell in Table 1, Table 3, Table 2, and the Wilson-CI tables from cached drafter outputs (released alongside the dataset).

For the specialised-prover column, the pinned GPTQ-quantised DSPv2 weights serve as the baseline endpoint.

Asset licenses. Benchmark JSONL, tier metadata, per-drafter closure records at $N = 5$, drafter prompts, 3-column hint ablation records, and the 14-session Aristotle history archive are released under CC-BY-4.0. The underlying Lean 4 library is released under Apache-2.0. The GPTQ-quantised DSPv2 weights are released as an artifact. Anonymised asset URLs are included in the supplementary archive.

Wilson 95% confidence intervals on Table 1. Table 7 reports the Wilson 95% confidence interval on every per-cell pass@5 closure rate from the headline matrix. The effective N for each Wilson interval is the per-tier denominator because pass@5 collapses each target's 5 attempts into a single Bernoulli outcome (target-closed if ≥ 1 of 5 attempts closed). The intervals are widest on the $n = 1$ T5 cell and the small-n T0, T2, T3, T4 cells (all per-tier denominator ≤ 8). The headline findings are stable under the Wilson CIs. The four-drafter T3 wall has every drafter $\leq 17\%$ closure CI upper bound, separated from Aristotle's 6/6 closure on the same column. Mistral's T1 collapse 1/34 has CI $[1, 15]\%$, below the 34 to 66% band of the other three drafters. The four-drafter T0 floor at 0/7 has Wilson upper bound 35% across all four, consistent with the Mathlib-canon framing.

I.1. $N = 10$ extension

To stress the headline closure pattern under a higher per-cell attempt budget, we extended all six drafter sweeps from $N = 5$ to $N = 10$ by firing five additional independent $T = 0.7$ samples per (target, drafter) cell, for a total of one greedy plus nine $T = 0.7$ attempts per cell. The slate, prompts, verification regime, and v2-strict sorry-detection are held identical to the $N = 5$ headline. The four closed-source drafters (Sonnet, Kimi, Gemini, Mistral), the supplementary fifth row (Opus), and the specialised prover (DSPv2-7B GPTQ-int8) were each extended on the full 60-target slate.

Across the $6 \times 60 = 360$ (drafter, target) cells, doubling the attempt budget shifts only *three* target-level closures: Gemini closes one new T2 target (`dichotomy_universal_monotonicity_impossible`), Opus closes one new T3 target (`etahr_over_etabetting_gt_one`), and DSPv2 closes one new T3 target (`etahr_le_etavector`). All three new closures are axiom-clean and pass v2-strict (no `sorry` in the candidate body). The remaining 285 cells reproduce their $N = 5$ verdict.

Table 8 reports the per-cell Wilson 95% CIs at $N = 10$.

Table 7. Wilson 95% confidence intervals on every per-cell pass@5 closure rate from Table 1. Cells report targets-closed/tier-size with the Wilson 95% CI in brackets (percent units). Mistral T1 1/34 at $[1, 15]$ is separated from the 34–66% band of Sonnet/Kimi/Gemini. The four-drafter T3 wall has every drafter at most 17% upper bound, below Aristotle's 6/6 T3 closure (§4.1). T4 and T5 cells report 0-imputed outcomes. Intervals reflect sampling uncertainty on the structural zero.

Drafter	T0 ($n = 7$)	T1 ($n = 34$)	T2 ($n = 8$)	T3 ($n = 6$)	T4 ($n = 4$)	T5 ($n = 1$)
Claude Sonnet 4.6	0/7 [0,35]	18/34 [37,69]	0/8 [0,32]	0/6 [0,39]	0/4 [0,49]	0/1 [0,79]
Kimi K2.5	0/7 [0,35]	17/34 [34,66]	0/8 [0,32]	1/6 [3,56]	0/4 [0,49]	0/1 [0,79]
Gemini 3 Pro	0/7 [0,35]	22/34 [48,79]	0/8 [0,32]	0/6 [0,39]	0/4 [0,49]	0/1 [0,79]
Mistral Large 3	0/7 [0,35]	1/34 [1,15]	0/8 [0,32]	1/6 [3,56]	0/4 [0,49]	0/1 [0,79]
Claude Opus 4.6	0/7 [0,35]	13/34 [24,55]	0/8 [0,32]	0/6 [0,39]	0/4 [0,49]	0/1 [0,79]
DSPv2-7B GPTQ-int8	3/7 [16,75]	8/34 [12,40]	0/8 [0,32]	1/6 [3,56]	0/4 [0,49]	0/1 [0,79]

Intervals widen only for cells whose target-level outcome changed. The 285 unchanged cells retain identical intervals to their $N = 5$ counterparts in Table 7. The headline findings hold under the $N = 10$ stress. Mistral T1 collapse remains at 1/34 with CI $[1, 15]$%. The four-drafter T3 wall has every drafter at most 17% upper bound (Kimi, Mistral, and Opus all at $1/6 = 17\%$ point estimate, with Sonnet and Gemini at 0/6). The four-drafter T0 floor remains at 0/7 for all four with Wilson upper bound 35%.

DSPv2 inference-path note. The DSPv2 $N = 10$ extension required switching inference paths (the original loader was unavailable at extension time). The first $N = 10$ attempt on the substitute path returned candidates with interior whitespace stripped ("`:=byintros...rw[quantizeReals]`" in place of "`:= by intros...rw [quantizeReals]`"). We diagnosed this as a chat-style-instruction prompt fed to a Lean-autocompletion model (DSPv2 is a completion model, not a chat model) and switched to a Lean-autocomplete prompt format (`{imports}\n\n{theorem_header} :=`). Re-running the sweep with the autocomplete prompt produced clean Lean candidates and one new T3 closure (`etahr_le_etavector`). A subsequent SDK release adds a model-family registry that auto-routes recognized Lean prover identifiers to the autocomplete format and warns loudly when a chat-style template is supplied for a known completion model. Researchers reproducing the DSPv2 sweep through the released SDK will get the autocomplete format by default without any caller-side ceremony.

Reproducibility note: trace-on-default for sweep launches. The drafter sweep drivers stream every `litellm.completion` event to a relational event log by default when the standard event-sink environment variables are set, allowing post-hoc per-target audit, cost rollup, and run-level dashboards. The default-on behaviour is implemented via a small wrapper that each driver's `main()` entry invokes. Passing `--no-trace` on the CLI skips the trace session entirely (useful for offline runs without a database sink). Researchers reproducing the headline sweep should rely on this default. Per-attempt JSON logs under `logs/` are written unconditionally and serve as the reconstruction path if a run was launched without the database sink available.

J. Refutation case study: Aristotle counterexample

Session `d2755ea2` submitted a candidate theorem asserting that the Gaussian small-ball lower bound holds without a symmetry hypothesis on the underlying filtration. Aristotle refuted the claim by constructing a two-step filtration witness in which the asserted bound fails by a constant factor. The witness was reduced to a Lean term and verified by `lake build`. The refuted statement was corrected by re-introducing the dropped symmetry hypothesis and re-submitted as session `54614669`, which closed cleanly. Both sessions and the counterexample witness are preserved in the history archive.

K. Extended capability analysis

This appendix supplies the four subsections trimmed from the main results in order to keep the main text within the page budget. The findings are supporting evidence for the main-text headline (Aristotle 6/6 T3 against drafter aggregate 2/24). The main text's §4 retains the headline findings.

K.1. Aristotle history archive

The companion archive of fourteen Aristotle sessions records:

- **Ten fully-closed sessions** with clean axiom audit {`propext`, `Classical.choice`, `Quot.sound`},

Table 8. Per-cell pass@10 closure rates with Wilson 95% CIs. T4 and T5 omitted (no drafter closes either at any N). Three cells differ from pass@5 in Table 7: Gemini T2 0/8 → 1/8, Opus T3 0/6 → 1/6, DSPv2 T3 1/6 → 2/6. Per-cell intervals are stable across $N = 5 \to N = 10$.

Drafter	T0 ($n = 7$)	T1 ($n = 34$)	T2 ($n = 8$)	T3 ($n = 6$)
Claude Sonnet 4.6	0/7 [0,35]	18/34 [37,69]	0/8 [0,32]	0/6 [0,39]
Kimi K2.5	0/7 [0,35]	17/34 [34,66]	0/8 [0,32]	1/6 [3,56]
Gemini 3 Pro	0/7 [0,35]	22/34 [48,79]	1/8 [2,47]	0/6 [0,39]
Mistral Large 3	0/7 [0,35]	1/34 [1,15]	0/8 [0,32]	1/6 [3,56]
Claude Opus 4.6	0/7 [0,35]	13/34 [24,55]	0/8 [0,32]	1/6 [3,56]
DSPv2-7B GPTQ-int8	3/7 [16,75]	8/34 [12,40]	0/8 [0,32]	2/6 [10,70]

covering HR admissibility, betting CS admissibility, wealth-process martingale property, power-loss bound, Gaussian small-ball lower bound, the universal Φ-transform admissibility for all four families (HR, vector, aCS, and implicitly betting via the Kelly-optimality characterisation), matching sharp constants c_{HR} and c_{vector} with their ranking proof, and two information-theoretic bounds (quantized entropy and Kelly-optimal betting).

- **Three partial-closure sessions** where Aristotle closed the headline theorem modulo residual specification sorries (these are recorded with the explicit residual list).
- **One refutation session** where Aristotle returned a counterexample: the submitted claim required a symmetry hypothesis that had been dropped during IP-strip. Aristotle's counterexample witness was preserved as an auxiliary dataset entry and triggered a correction to the original statement.

The 14-session archive supplies ground-truth closures for a subset of benchmark targets, which allows drafter-side closures to be cross-validated against Aristotle's certified output.

K.2. Aristotle capability demonstrations beyond the main table

In addition to the benchmark-Table 1 evaluation, Harmonic Aristotle was exercised on three companion-archive theorems during library development: `c_sharp_ranking` (the sharp-constant ranking $c_{\mathrm{aCS}} \leq c_{\mathrm{vector}}$, a sharp-constant statement distinct from the rate-function ranking $\eta_{\mathrm{betting}} \leq \eta_{\mathrm{aCS}} \leq \eta_{\mathrm{HR}} \leq \eta_{\mathrm{vector}}$ that appears at T3 of the benchmark), `quantized_entropy_bound` (an information-theoretic entropy bound applied to the quantization-coarsened CS), and `betting_is_kelly_optimal` (the Kelly-optimality characterisation of the betting CS). All three closed axiom-clean. These are *not* Table 1 benchmark entries because the library proofs predate the evaluation. we report them to contextualise Aristotle's capability on theorems adjacent to the benchmark targets. No claim about Aristotle's benchmark tier close rate is made from these companion-archive closures.

K.3. Cross-drafter tactic-prior diversity

On four targets in the benchmark, *every* evaluated closed-source drafter and Aristotle both close. These dual-closure rows expose cross-drafter variance: different commercial drafters close the same theorem via structurally *different tactic chains*, and one drafter may close where another fails at identical attempt budget. A concrete example is the sharp-constant ranking `c_sharp_ranking` ($c_{\mathrm{aCS}} \leq c_{\mathrm{vector}}$): Kimi K2.5 fails 0/9 at the SUB hint (its tactic prior directs search toward `one_div_le_one_div_iff`-style linear-algebra routes that terminate without discharging the goal), whereas Gemini 3 Pro closes at att 1 greedy with the concise chain `[unfold, gcongr, linarith [Real.pi_pos]]`. The same target is also closed by Aristotle under a longer search, and each of the three closures uses a distinct tactic route. We interpret this as drafter-specific tactic priors: different drafters have structurally different views of the Mathlib lemma graph, and the benchmark's hint-ablation protocol exposes that variance as information rather than noise.

Convergent T3 close on the $\sqrt{2}$ identity. The single T3 target both Kimi and Mistral uniquely close at $N = 5$ greedy is `etavector_eq_sqrt_two_mul_etahr` ($\eta_{\mathrm{vector}} = \sqrt{2} \cdot \eta_{\mathrm{HR}}$). Kimi closes it on the greedy attempt and reproduces at two $T = 0.7$ attempts. Mistral (under the neutral prompt) closes at the greedy attempt and reproduces at one $T = 0.7$ attempt. Sonnet and Gemini miss this target at $N = 5$. Two independent commercial drafters converging on the same cross-family identity under 0 explicit hint is direct evidence that the $\sqrt{2}$ factor between the HR and Whitehouse-vector rate functions is the most accessible cross-family relation in the 4-family rate landscape, independent of

drafter tactic-prior.

K.4. Four solver regimes in the capability-wall tail

Running the full drafter sweep (Kimi K2.5, Sonnet 4.6, Gemini 3 Pro) against the five capability-wall tail theorems under the SUB and SUB+CHAIN hint regimes yields 5 closures in 100 total attempts (5.0% drafter close-rate). Breaking the closure matrix out by target reveals *four qualitatively distinct solver regimes* plus a deepest-tier target unreachable by any generalist drafter (Table 9):

The four regimes correspond directly to the capability axes identified in §4. *Universal* targets exercise only baseline tactic chains (well within every drafter's Mathlib prior). *Majority* targets exercise a rewrite step that most but not all drafters' priors reach. *Drafter-monopoly* targets expose cross-drafter tactic-prior divergence (Gemini's `gcongr` prior vs Kimi's `one_div_le_one_div_iff` prior). *Aristotle-monopoly* targets require a multi-step lemma chain that all generalist drafter priors miss. The *specialised-only tier* shows the tactic-vocabulary ceiling. Drafters have the mathematics but lack the Lean 4 API signatures needed to close. The dataset records the closure matrix at row granularity, enabling per-target regime analyses.

K.5. Axiom audit

Every closed Lean 4 proof in the release terminates with a `#print axioms` check. The canonical axiom set for a classical Mathlib proof is $\mathcal{A}_{\text{core}} := \{\texttt{propext}, \texttt{Classical.choice}, \texttt{Quot.sound}\}$. The axiom-set union across all closed proofs in the release is exactly $\mathcal{A}_{\text{core}}$, no proof widens the axiom set beyond this baseline (Table 10).

Why this matters. A capability benchmark that accepts proofs widening the axiom set would confound tactic-search capability with axiom-cheat pathology. Requiring axiom-set stability across all closed proofs (enforced by the drafter sweep's closure criterion at §3) makes the benchmark's closure counts measurements of *constructive* proof-search capability under a fixed logical foundation.

K.6. Failure-mode taxonomy of the drafter sweep

Each drafter attempt resolves into one of five outcomes, summarised in Table 11. *Closed* attempts type-check, satisfy the goal modulo no extra sorries, and do not widen the axiom set beyond $\mathcal{A}_{\text{core}}$. *Drafter-wrote-sorry* attempts emit at least one `sorry` token in the candidate proof body, signalling explicit drafter abstention. *Type-check failure* attempts emit a candidate that the Lean elaborator rejects (unknown identifier, type mismatch, unsolved goal, or tactic failure). This is the dominant failure mode at $\sim 80\%$ of attempts and is the conventional capability-blocked regime. *Sample-side error* attempts return no usable proof body (API timeout, empty completion, format violation). These are sampling-pipeline failures rather than capability signals. *Alt-proof non-close* attempts return a candidate distinct from the reference Aristotle proof that nevertheless does not type-check. This category is rare ($< 0.5\%$).

K.7. Latency and cost profile

Table 12 reports per-drafter wall-clock and per-attempt API cost aggregated across the 60-target benchmark drafter sweeps (one greedy attempt at $T = 0$ plus four $T = 0.7$ independent samples, per target). The cost column is the per-attempt USD cost at Anthropic/Gemini/Mistral list prices. Wall-clock is the full round-trip including network latency.

Cost per closed target. We report API cost per closed target, defined as the per-drafter total API cost on the 60-target benchmark divided by the number of targets closed at pass@5. This avoids treating attempts as independent Bernoulli trials, which they are not. On the 60-target benchmark at pass@5: Mistral closes 2/60 at \$0.87 total (\$0.44 per closed target on the neutral prompt). Sonnet closes 18/60 at \$5.71 total (\$0.32 per closed target). Kimi closes 18/60 at \$2.30 total (\$0.13 per closed target). Gemini closes 22/60 at \$14.67 total (\$0.67 per closed target). Opus closes 13/60 at \$13.75 total (\$1.06 per closed target). DSPv2 closes 12/60 at \$0 marginal API cost. The ranking changes if measured per attempt rather than per closed target, because attempt cost and closure are not linearly related. We do not derive a per-attempt expected closure rate.

K.8. Glossary of formal-methods terms

Anytime-valid CS. A confidence sequence $(C_t)_{t\geq 1}$ at level $1-\alpha$ is *anytime-valid* when $\mathbb{P}(\forall t \geq 1 : \mu \in C_t) \geq 1-\alpha$. The guarantee holds at every stopping time simultaneously, fixing optional stopping in sequential experiments.

Rate function η_F. Per-family CS half-width function. Smaller η_F means a tighter CS. Cross-family inequalities of the form $\eta_F \leq \eta_G$ rank the families.

Φ-substitution. Log-domain substitution $x \mapsto \exp(\lambda^\star x - (\lambda^\star)^2\sigma^2/2)$ that converts an additive cu-

Table 9. Four solver regimes in the no-closure tail. Cells are (close count)/(attempt count) at $N \leq 9$ per drafter. †Solver-unreached: no closure across (a) 3 drafters $\times N = 5$ greedy unhinted, (b) 3 drafters $\times$ asymmetric SUB, (c) 3 drafters $\times N = 5$ SUB+CHAIN, (d) 3 Aristotle attempts. ‡Two Aristotle sessions timed out at 20 min. One returned a counterexample on the as-stated form (§4.4). We avoid claiming *fundamentally* unreachable.

Target	Kimi	Sonnet	Gemini	Aristotle	Regime
`betting_is_kelly_optimal`	1/1	1/1	1/1	1/1	**universal**
`c_vector_eq_sqrt_two_mul_c_HR`	1/1	0/5	1/2	1/1	**majority**
`c_sharp_ranking`	0/9	0/5	1/3	1/1	**drafter-monopoly**
`quantized_entropy_bound`	0/9	0/5	0/3	1/1	**Aristotle-monopoly**
`equivalence_break_at_finite_precision`	0/19	0/15	0/13	0/3 (timeout‡)	**solver-unreached**†

Table 10. Axiom audit. Every closed proof uses exactly $\mathcal{A}_{\text{core}} = \{$`propext`, `Classical.choice`, `Quot.sound`$\}$. Refuted targets return a counterexample witness, no proof. Three refutations are documented in §4.4 and §J. No closed proof widens the axiom set beyond $\mathcal{A}_{\text{core}}$ on either side.

Source	Verdict	N	Axiom set
Aristotle main table (T0/T1/T2)	closed	47	$\mathcal{A}_{\text{core}}$
Aristotle main table (T1/T2)	refuted	2	(counterexample, no proof)
Aristotle main table (T3)	closed	6	$\mathcal{A}_{\text{core}}$
Aristotle main table (T5)	refuted	0	(counterexample, no proof)
Aristotle main table (T4)	attempted	0	library-gap, no proof
Aristotle companion archive	closed	3	$\mathcal{A}_{\text{core}}$
Aristotle companion archive	refuted	1	(counterexample, no proof)
Drafter closures (all)	closed	≥ 1 per closed row	$\mathcal{A}_{\text{core}}$
Release axiom-set union	n/a	n/a	$\mathcal{A}_{\text{core}}$

Table 11. Per-drafter and aggregate failure-mode breakdown of the $N = 5$ sweep across T0/T1/T2 (T3 ablation columns in Table 3). The dominant non-close mode is type-check failure. Opus's sorry rate is $52/480 = 10.8\%$, an order of magnitude above Sonnet's 2.4% and the other three drafters' $\leq 0.2\%$. Kimi and Mistral each have 85 sample-side errors concentrated on T2 (truncated or empty API completions), an integration artefact.

Drafter	closed	sorry	type-fail	sample-err	alt-noclose	total
Sonnet 4.6	50	14	530	0	1	595
Kimi K2.5	100	0	375	85	5	565
Gemini 3 Pro	103	1	461	0	0	565
Mistral L 3	8	1	471	85	0	565
Opus 4.6	61	52	365	0	2	480
aggregate	322 (11.6%)	68 (2.5%)	2202 (79.5%)	170 (6.1%)	8 (0.3%)	2770

Table 12. Drafter sweep latency and cost profile on the 60-target benchmark. Attempt counts include duplicate runs after a tokenizer fix. Per-attempt cost uses $T = 0$ greedy plus $T = 0.7$ samples at list prices. Total API-priced cost across the five closed-source drafters is \$37.30. DSPv2 and Goedel-V2 run locally with no API cost. Token counts are approximate.

Drafter	attempts	per-att wall (sec)	per-att cost (USD)	tok-in (avg)	tok-out (avg)	total (USD)
Claude Sonnet 4.6	821	7.9	\$0.0070	160	432	\$5.71
Kimi K2.5	480	23.1	\$0.0048	157	4400	\$2.30
Gemini 3 Pro	480	29.3	\$0.0306	317	3017	\$14.67
Mistral Large 3	506	3.2	\$0.0017	166	233	\$0.87
Claude Opus 4.6	480	8.2	\$0.0287	203	341	\$13.75
DSPv2-7B (community GPTQ-int8)	240	3.2	\$0	∼190	∼130	\$0
Goedel-V2-8B (community GGUF Q6_K)	240	21.8	\$0	111	535	\$0

mulant inequality to a multiplicative exponential supermartingale. Central to the cross-family ranking proofs.

Lateral re-encoding. Capability of proposing the Φ-substitution autonomously from the goal statement without explicit prompt guidance.

Tactic vocabulary. Capability of selecting the sequence of Mathlib lemmas (`Real.sqrt_le_sqrt`, `nlinarith`, etc.) that closes a re-encoded goal.

Axiom-clean closure. A Lean kernel certificate where `#print axioms` returns exactly {`propext`, `Classical.choice`, `Quot.sound`}. The canonical Mathlib core.

MCTS. Monte Carlo Tree Search, a search policy used by some specialised theorem provers to explore tactic sequences. Aristotle's internal search policy is not publicly documented.

Optional stopping. Choosing the sample size based on the data already observed. Inflates classical fixed-horizon type-I error from the nominal α to several times higher in practice.

pass@N. A target is closed at pass@N if at least one of N independent attempts produces an axiom-clean Lean kernel certificate.

K.9. How to extend the benchmark

The benchmark's tier structure is designed to admit new targets at any of T0 through T3, plus the T4/T5 frontier. To submit a new target:

1. Author the theorem statement in a Lean module under `formal-avs-lean/FormalAvs/`, with the appropriate imports.
2. Assign a paper tier: T0 (Mathlib-canon), T1 (single-function friendly), T2 (single-function challenging), T3 (cross-function), T4 (Mathlib-gap), T5 (capability-ceiling).
3. Add a JSONL row to the released `benchmark.jsonl` with: `benchmark_id`, `statement_lean`, `tier`, `required_imports`, `requires_formal_avs_lean`, and the appropriate `drafter_close_rate_*` fields (initially `null`).
4. Run the drafter sweep at $N = 5$ via the released `sweep.py` against the four headline drafters and DSPv2. Populate the `drafter_close_rate_*` fields.
5. For T3 targets, additionally provide the Sub and Sub+Chain hint columns and run the 3-column ablation.
6. For Aristotle coverage, submit the target as a session via the public Aristotle API. Record session_id, verdict, and axiom audit in `aristotle_history.jsonl`.
7. Open a pull request against the public benchmark repository.

The benchmark is versioned. Snapshot tags (`formal-avs-v1.0`, `v1.1`, ...) freeze the target set, drafter close-rate columns, and Aristotle history at a specific commit so reported results stay reproducible.

K.10. Aristotle session metadata

The 14-session companion archive records, per session, fingerprint, target-set, verdict, axiom-clean status, and elapsed wall time. Table 13 summarises.

K.11. T0 stripped-prompt disambiguation

The headline T0 close-rate of 0/28 across the four generalist drafters could plausibly be explained either by a tactic-prior or retrieval gap on Mathlib-canon lemmas, or by an artefact of the hint-list prompt template (the banned-tactics clause, the allowed-tactics enumeration, or the namespace preamble suppressing otherwise-feasible closures). To disambiguate, we pre-registered and ran a stripped-prompt control: theorem statement only, no banned-tactics list, no allowed-tactics enumeration, no namespace preamble, on 6 of the 7 T0 targets across the four drafters at $N = 5$ (120 attempts total). The stripped prompt closed 0/120, matching the hint-list rate. This rules out the prompt-wrapping hypothesis. The remaining T0 target (`div_le_div_of_nonneg_left`) was excluded because its as-stated form was refuted by Aristotle (§4.4). The corrected form is a fair test for a future revision.

K.12. N=25 T3 replicate

To shrink the wide 95% Wilson CI on the headline T3 aggregate 2/24, we ran a replicate sweep at $N = 25$ on the six T3 cross-function targets across the four generalist drafters (Sonnet 4.6, Kimi K2.5, Gemini 3 Pro, Mistral Large 3) for a total of $4 \times 6 \times 25 = 600$ attempts. Five of the six T3 targets reconstruct cleanly into the sweep driver. One (`etahr_over_etabetting_gt_one`) fails statement reconstruction at $N = 25$ and is excluded.

The unit of analysis for pass@N is the target-drafter cell (§3, drafter sweep protocol), not the individual attempt. We therefore report cell-level pass@25 rather than aggregating attempts as if independent.

Table 13. Aristotle 14-session archive summary. All "closed" rows are axiom-clean ($\mathcal{A}_{\text{core}}$). † companion-archive sessions (§K.2). ‡ library-development sessions whose targets are not in the benchmark. Full per-session JSON in `aristotle_history.jsonl`.

Session id	Scope	Verdict	Wall (min)
`74fe5c49`	6 T3 cross-function	closed	42
`f0bac40c`	48-slate (T0/T1/T2)	46 closed + 2 refuted	51
`76c5d525`	7 paper additions	closed	36
`bd3ea302`	4 T4 library-gap	no proof	50
`5ac0658d`	1 T5 capability-ceiling	refuted (symmetry)	44
`d2755ea2`	Gaussian small-ball	refuted (symmetry)	20
`54614669`	corrected small-ball	closed	14
companion† (3 sessions)	sharp-rank, entropy, Kelly	all closed	~ 30 each
4 additional‡	library-development	mixed	varies

Drafter	T3 pass@25 (5 targets, neutral prompt)
Claude Sonnet 4.6	0/5
Kimi K2.5	0/5
Gemini 3 Pro	0/5
Mistral Large 3	0/5
Aggregate (4 drafters $\times$ 5 targets)	0/20 cells

The cell-level Wilson 95% upper bound on pass@25 at 0/20 is approximately 16.1%. This is evidence on the no-hint regime on 5 of the 6 T3 targets. It does not directly bound the headline hint-list result (2/24 at $N = 5$ across 6 T3 targets), since the prompts and target sets differ. The total compute spent on this replicate ($4 \times 5 \times 25 = 500$ attempts) is reported under §K.7 for transparency, but is not used to compute an attempt-level pass rate.

L. Prompt-sensitivity detail

The hint-list prompt improves three of four generalist drafters and reduces Mistral's closure rate.

Mistral Large 3. The hint-list acts as an action-space restriction. Mistral's hint-list attempts converge to a deterministic `:= by positivity` pattern, selecting one tactic from the enumerated list (0/60 on hint-list). Without the enumeration, pass@5 rises to 2/60, recovering the T3 $\sqrt{2}$ identity and one T1 target.

Sonnet, Kimi, Gemini. The hint-list provides a useful prior. Removing it reduces pass@5 by a drafter-specific factor: Sonnet 18 $\rightarrow$ 3 (6$\times$), Kimi 18 $\rightarrow$ 13 (1.4$\times$), Gemini 22 $\rightarrow$ 13 (1.7$\times$). Sonnet's drop is not a parsing artifact: 88.6% of neutral attempts contain a parseable Lean block. The prompt-sensitivity result argues against reading the headline table as a single leaderboard.

M. Inference-format analysis

On our slate, DSPv2-7B (Ren et al., 2025) closes 9/60 at pass@5 while Goedel-Prover-V2-8B (Goedel-LM Authors, 2025) closes 5/60 under single-shot, despite Goedel-V2 reporting higher closure rates on published benchmarks (MiniF2F, ProofNet). Four of five Goedel-V2 closures overlap with DSPv2. The gap is a hardware-constraint artifact: DSPv2 outputs pure Lean in 50–80 tokens (autocomplete format), while Goedel-V2 is chat-trained with a recommended 32k-token inference window. On a 10 GB consumer GPU the maximum context is 4096 tokens (12.5% of the recommended budget).

Under agentic mode (`cycle_prove` r=4), Goedel-V2's verifier-guided self-correction training activates: the model uses Lean compiler feedback to iteratively revise its proofs, rising from 5/60 to 13/60 (2.6$\times$). This surpasses DSPv2's single-shot 9/60, making Goedel-V2 the strongest local prover on our slate at \$0 marginal cost. The self-correction capability is the differentiator: DSPv2 (autocomplete-trained, no self-correction) cannot benefit from a refinement loop.