AUDITING CHAIN-OF-THOUGHT REGISTER EFFECTS IN LEAN 4 PROOF GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Does the register of a chain-of-thought (CoT) prompt matter when a language model writes Lean 4 proofs? We evaluate five prompt conditions on 11 models and 21,328 analyzed model–condition–problem cells from a 500-problem LeanWorkbook draw, and report two findings that appear only once proof scoring is itself audited. First, the scoring pipeline carried two defects: the verifier read Lean's `sorry` warning from the wrong stream, counting unfinished proofs as successes, and the extractor handed Lean text the model never offered as its proof. Screening every stored success removes 204 that had closed with `sorry`, and re-verifying the 6,548 cells whose proof text the repair changes recovers 764 real ones (panel successes 1,696→2,256). A further 1,205 cells were truncated before storage and cannot be re-scored; we retain their old verdicts and bound the uncertainty this creates; the adversarial bound includes no effect. Second, on the re-scored panel NL CoT alone has a panel-average interval excluding 1 (OR 1.58, 95% HDI, a highest-density interval, $[1.14, 2.10]$; raw panel rates 8.4%→10.6%), while Lean pseudocode, symbolic math and tactic plans point above 1 with intervals containing it. The average does not determine the effect for an untested model: heterogeneity persists (τ 0.17–0.47) and the predicted interval for a new model spans 1 in every register. After re-scoring, neither self-rewriting nor an independent renderer establishes a pooled presentation effect (OR 1.14 $[0.91, 1.43]$ and 1.02 $[0.86, 1.21]$); with five samples per problem on two models, no register's paired gain over Direct excludes zero. We report a small average benefit for one register; these data do not establish a prompting rule for an untested model.

1 INTRODUCTION

Fine-tuned neural theorem provers perform strongly (Ren et al., 2025; Hubert et al., 2025; Chen et al., 2025), yet generic LLMs still struggle to produce proofs that a formal checker accepts (Yang et al., 2023). Chain-of-thought (CoT) prompting (Wei et al., 2022), which helps on arithmetic and commonsense reasoning, is the obvious first remedy, and in theorem proving it raises a question those tasks do not: in which *register* should the model reason before it writes Lean? The reasoning could be a natural-language (NL) sketch, Lean pseudocode, symbolic math, or tactic plans. We ask whether that choice has a stable answer across models and, where it does not, whether anything measurable explains the variation.

We evaluate 11 models spanning six families, with standard and reasoning post-training, on the same 500 LeanWorkbook problems under five conditions: direct generation, NL CoT, and three *formal registers* (Lean pseudocode, symbolic math, tactic plans), every proof checked by the Lean compiler (Figure 1). The answer has an average and a boundary. NL CoT has a *modest positive panel-average effect* and is the only register surviving every level of pooling, but that average licenses no prompting rule: model identity shifts a register's effect enough to change its sign; pre-audit probes suggest collection window may too. Nor is "formal CoT" one treatment — the three formal registers behave differently from each other, and the harm the uncorrected pipeline showed in them was an artifact of proof extraction, not a property of the models.

The rest of the paper asks what would turn this average into a rule. A mechanism could explain it: register-mismatched context interfering with the proof distribution (§5), or the register in which reasoning is presented rather than its authored content (§6). Or the effect could depend on some-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

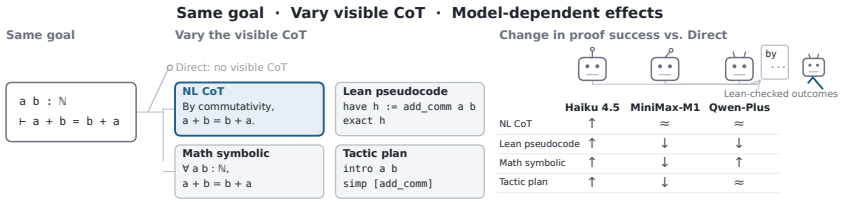


Figure 1: **Same proof task, different reasoning registers, model-dependent outcomes.** For each theorem we compare Direct generation with four visible-CoT conditions and verify generated proofs with Lean. Arrows show selected models' aggregate point-estimate directions relative to their own Direct, not statistical significance, and $\approx$ marks a gap of at most four solved problems; the goal shown is illustrative. Primary pooled effects appear in Table 2; per-model estimates appear in Figure S1.

thing observable, such as the provider's reasoning label (§7) or the benchmark (§7). The register-distance association did not predict held-out ordering, and after re-scoring neither the planned self-rewriting analysis nor the independent-renderer replication establishes a pooled presentation effect. The provider label has unresolved moderation under wide bounds, and the second benchmark lacks power to establish how its effects compare (§7).

Our contributions:

1. **A scoring audit that changes the answer** (§3). Two defects — a verifier reading Lean's `sorry` warning from the wrong stream, and an extractor compiling text the model never offered as its proof — did not cancel across conditions. Re-scoring removes 204 spurious successes and recovers 764 real ones, reversing the formal-register harm on identical data (Table 1).
2. **A measurement of CoT register effects** across 11 models with compiler-verified outcomes at three levels of pooling (§4): NL CoT has a modest within-panel average benefit, while the formal registers do not act as a single condition and their average-effect intervals leave harm unresolved (Table 2).
3. **Evidence that this average is not a transferable rule** (§4.1, §7): effects vary by model; four labeled endpoints cannot resolve reasoning-label moderation; and a single-window second benchmark is too underpowered after re-scoring to say whether the ordering transfers.
4. **Two mechanism candidates tested beyond their design sets** (§5, §6): interference ordering, teacher-forced on a fixed gold proof, with each model's *register distance* in sample, and a presentation effect under self-rewriting. Pre-registered replication confirms neither; after re-scoring the presentation effect fails in its own design too.

2 Framework

We model CoT as a latent variable in proof generation (Xie et al., 2022): given theorem x, proof y, trace z and model parameters θ, $p_\theta(y|x) = \sum_z p_\theta(y|z,x)\, p_\theta(z|x)$, with $\mathcal{Z} = \mathcal{Z}_{NL} \cup \mathcal{Z}_F$ partitioned by modality.

The two hypotheses below are interpretive. Each states conditions under which we would expect CoT to hurt; the experiments test them rather than presuppose them, and the paragraph after them summarizes which parts the data support.

Hypothesis 2.1 (CoT Modality Harm)**.** *Let $A_m = \mathbb{E}_{z\sim p_\theta(z|x), z\in\mathcal{Z}_m}[p_\theta(y^*|z,x)]$. We expect $A_F < A_{direct}$ for a given model to the extent that:* ***(C1)*** *its formal CoT distribution is misaligned with its proof-competent distribution;* ***(C2)*** *formal verification is discontinuous (near-miss proofs score zero);* ***(C3)*** *conditioning on one externally sampled trace commits the model to that trace, whereas direct generation avoids such a commitment. C3 concerns what conditioning forecloses; it does not claim that the model performs an explicit marginalization. Because C1 is a* model property*, this hypothesis predicts heterogeneous rather than uniform effects.*

Hypothesis 2.2 (External-CoT Interference for Reasoning Models)**.** *For models whose internal reasoning distribution $p_\theta^{RL}(z_{int}|x)$ was optimized by RL, an external CoT prefix z_{ext} shifts internal reasoning toward its register and away from the optimized strategy, predicting $A_{ext} \leq A_{direct}$ when the shift outweighs any informational benefit. This has content only insofar as the shift is register-directed and accuracy-relevant, which requires model-internal measurement.*

What the data say. Hypothesis 2.1 predicts heterogeneous effects, and the panel confirms that (§4). Its *directional* prediction $A_F < A_{direct}$ is *not* supported once scoring is corrected: all three formal registers point above 1 with intervals containing it, the pre-audit harm having been an extraction artifact (§3). The weaker half survives: register effects are heterogeneous. The corrected primary analysis does not establish average harm for any formal register, and NL CoT alone has a 95% interval entirely above the null, though neither ordering holds for every model. Hypothesis 2.2 finds no support as a label-level claim (§7): the sample contains counterexamples in both directions, and label moderation remains unresolved.

Related Work. CoT prompting (Wei et al., 2022; Sprague et al., 2025) often helps mathematical reasoning, but recent work identifies failure modes: brittleness beyond training distributions (Zhao et al., 2025), sensitivity to prompt format and intermediate language (Sclar et al., 2024; Beiser et al., 2025), and harm on tasks where deliberation also hurts humans (Liu et al., 2024). In theorem proving, NL-to-formal pipelines outperform direct formal generation (Jiang et al., 2023; Varambally et al., 2025), and the DeepSeek-R1 report (DeepSeek-AI, 2025) documents few-shot prompting overriding internalized reasoning. Shojaee et al. (2025) and Wang et al. (2025) study the limits of thinking models but not how external and internal CoT interact. These studies do not measure how the register of an external CoT interacts with a compiler-verified proof pipeline across models, which is the gap this paper addresses; nor, as we show, do they audit how the proofs themselves are scored.

3 How Proofs Are Scored

Every number in this paper is a compiler verdict, so the pipeline that turns a model's text into a verdict is part of the measurement. Auditing ours changed the paper's conclusion, and we report the audit first because it conditions everything after it.

Two defects. *(1) Unfinished proofs counted as successes.* Lean writes `declaration uses 'sorry'` to standard output; our verifier tested standard error, which carries only the build tool's own notices. A proof closing its goal with `sorry` or `admit` therefore compiled to a "success." *(2) Lean was given text the model never offered as its proof.* Our extractor took everything after the first `-- Proof:` marker. When a model restated the theorem header before its tactics, echoed the few-shot exemplars before answering, or appended prose after a code fence, all of that reached the compiler. A related defect flattened indentation, which breaks nested `have ... := by` and `calc` blocks.

Scale and direction. Of the $21{,}328$ analyzed cells, the repaired extractor changes the proof text in $6{,}548$ (30.7%); we re-verified each of those in Lean 4. Correction removes 204 spurious successes and recovers 764 real ones, moving panel successes from $1{,}696$ to $2{,}256$. Neither defect is register-neutral: restated headers and echoed exemplars fell almost entirely on the formal registers (890 failing cells opened with a theorem declaration; zero under *Direct*), while the `sorry` inflation ran the other way, concentrated in *Direct* and NL CoT (42 of 354 and 71 of 389 stored successes). Together they manufactured the formal-register harm and inflated part of NL's advantage.

What cannot be recovered. $1{,}205$ cells were truncated at a response-storage cap; their corrected verdicts are unknown, so we retain the stored verdicts and bound the effect of that choice in §8. Another 857 yield no extractable tactics and are determinate failures under the repaired rule: $2{,}062$ cells in all cannot be retrospectively compiled. Two endpoints leave the panel entirely: DeepSeek-R1, whose cells store no response text and whose one re-collection returned empty completions for 39–43% of cells, and Kimi-K2-Thinking, excluded for unrepairable API failures. Verifier timeouts also orphaned compiler processes, so stored wall-clock times are unreliable; no verdict depends on them.

Table 1: **What the audit changed**, on identical data: the same 21,328 cells and the same 11 models, refit with only the verdict rule changed (common-slope mixed-effects logistic model, or GLMM, odds ratio vs. *Direct*, 95% CI). Three significant formal-register harms before correction; none after. Generated by `scripts/before_after_audit.py`.

Verdict rule	NL CoT	Lean pseudocode	Math symbolic	Tactic plan
Before (stored)	1.29 [1.14, 1.47]	0.69 [0.60, 0.79]	0.81 [0.71, 0.92]	0.68 [0.59, 0.78]
After (corrected)	1.69 [1.48, 1.93]	1.50 [1.33, 1.70]	1.63 [1.44, 1.84]	1.18 [1.04, 1.34]
Success rate	9.8→10.6%	7.0→11.6%	7.8→12.0%	6.9→10.2%

Direct moves 8.5→8.4%; panel successes 1,696→2,256.

Controls, and what the audit covers. The defects act in opposite directions and need different coverage. A `sorry` can only turn a real failure into a recorded *success*, so screening every stored success audits it completely rather than by sample: all 1,696 stored successes of the eleven analyzed models were re-verified in Lean — every arm at 100%, *Direct* included — and 204 were `sorry`-closed. The extraction defect acts both ways, so it is audited over the 6,548 cells whose proof text the repair changes. The repaired verifier carries a regression test, and the recoveries are audited as a census: in all 764, the extracted tactic lines appear *verbatim* in the model's own response, so the repair never fabricated proof text — though a block lifted from a Lean-like CoT need not be what the model *offered* (Appendix B). The verifier fixes each target statement before inserting tactics; a restated weaker theorem inside the tactic block cannot replace the target.

4 The Average Register Effect

The five prompt instructions and answer slots are summarized in Appendix A; the exact generators are supplied with the paper.

We evaluate 11 LLMs on 500 problems from LeanWorkbook (Ying et al., 2024) under five CoT variants (direct, NL, tactic plan, math symbolic, Lean pseudocode), with all proofs verified by the Lean 4 compiler.[1] Before any LLM call, a baseline of 16 single tactics (including `aesop`, `omega`, `nlinarith`, `polyrith` and `decide`) solves 98 problems. Unless noted, analyses use the remaining LLM-contribution regime, with binary compiler success as a thresholded proxy for the proof-competence quantity A_m of Hypothesis 2.1. A stronger 31-operator baseline absorbs a further 1.7% of the regime and moves every register odds ratio by ≤ 0.012 on the re-scored data, so an understated baseline does not drive the results (Appendix J). Appendix B gives all three success-rate definitions per cell, per-model counts, and the parser audit.

One panel, one collection window. Repairing an earlier data-integrity fault meant re-running some cells months later, so the panel holds every model to a single collection window (§4.1); DeepSeek-R1, the one endpoint that would have needed cross-window reconstruction, is excluded for the unrelated reason that it cannot be re-scored (§3).

Three levels of pooling. A model's raw success difference can be noisy, so we analyze the 11 models under three assumptions about how effects vary: the primary fit is Table 2, and the other two levels, with per-model estimates, are in Appendix C. An odds ratio (OR) above one favors the named register over Direct; an interval spanning one leaves its direction unresolved.

(i) Common slope. A binomial mixed-effects logistic regression with random intercepts for problem and model and one fixed effect per variant gives OR 1.69 [1.48, 1.93] for NL CoT, 1.50 [1.33, 1.70] for Lean pseudocode, 1.63 [1.44, 1.84] for symbolic math and 1.18 [1.04, 1.34] for tactic plans: on re-scored data every register sits above 1 at this level of pooling. Because the slope is shared across

[1]The 11 endpoints: GPT-4o-mini, o3-mini (OpenAI); Claude Haiku 4.5 (Anthropic); Qwen-Plus, Qwen3-30B-A3B instruct and thinking (Qwen); DeepSeek-Chat, DeepSeek-V4-Pro, DeepSeek-V4-Flash (DeepSeek); GLM-4-Plus (Zhipu); MiniMax-M1 (MiniMax). Two are excluded whole-row on recoverability, not outcome, by criteria fixed beforehand: Kimi-K2-Thinking (27 API-failure cells, after which the endpoint was retired everywhere we could reach) and DeepSeek-R1 (no stored response text, so it cannot be re-scored (§3); its only re-collection returned empty completions for 39–43% of cells, so we report no R1-inclusive fit).

models, the estimate is a within-panel average. Refitting 14 times — dropping each of the 11 models and each of the 3 multi-model families — plus the full-panel fit gives 15 fits (Appendix C): NL CoT and symbolic math stay significant in all 15 (OR 1.44–1.80 and 1.33–1.83), Lean pseudocode in 14 of 15 and tactic plans in 11 of 15; the one fit breaking both drops the three DeepSeek models, which supply the largest share of re-scored cells.

(ii) Meta-analysis. For each model and register we count *discordant pairs*: problems solved by exactly one of that register and Direct. We pool their paired odds ratios with DerSimonian–Laird random effects (Appendix C); Hartung–Knapp (HK) widens uncertainty for only 11 models. NL CoT alone survives that correction (OR 2.05, HK CI $[1.25, 3.36]$, p=0.009). The other three keep positive point estimates but lose significance once each model has its own true effect (Lean pseudocode 1.36, p=0.16; symbolic math 1.44, p=0.21; tactic plans 1.22, p=0.30): between-model heterogeneity dominates, with I^2=45–76% and between-model SD τ=0.38–0.73 on the log-OR scale (a different estimand from the random-slopes τ in (iii)). Because I^2 alone cannot say whether that spread exceeds binomial noise, we test it: Cochran's Q on 10 df is 34.9 (p=1.3×10^{-4}, Lean pseudocode) and 41.6 (p=8.9×10^{-6}, symbolic math), both Holm-significant, but 19.4 (p=0.036, NL CoT) and 18.3 (p=0.051, tactic plans), neither surviving Holm and both losing significance if any single model is dropped. Heterogeneity beyond sampling noise is established for the two structured formal registers and borderline for the other two.

(iii) Random slopes (primary). Model (i) assumes one effect across models, and (ii) treats models sharing the same problems as independent studies. Our primary model allows each model its own register response: a Bayesian hierarchical logistic regression with per-model random *slopes* for each register and random intercepts for model and problem (NUTS, 4 chains $\times$ 1,000 draws after 1,000 warmup; $\hat{R}\leq1.01$, min bulk ESS 449, min tail ESS 844, 0 divergent transitions, BFMI 0.66–0.75). NL CoT has panel-average OR 1.58, 95% HDI $[1.14, 2.10]$ and $P(\text{OR}<1)$=0.003; it is the only register whose HDI excludes 1. The other three point above 1 with HDIs that contain it (Lean pseudocode 1.19 $[0.77, 1.69]$; symbolic math 1.37 $[0.83, 2.02]$; tactic plans 1.15 $[0.85, 1.52]$); their intervals do not establish either benefit or absence of harm. The between-model slope SD is τ=0.17–0.47. The model-based predictive interval treats these sampled endpoints as exchangeable with the untested one; it is not an empirical guarantee for every future API. For a practitioner the useful quantity is the posterior predictive for an *untested* model: every register's 95% prediction interval spans 1, including NL CoT's $[0.83, 2.83]$ with $P(\text{harm})$=0.06. That conclusion is not an artifact of the prior on τ: a prior-free random-effects prediction interval from (ii) is 1.6–2.0 times *wider* and also spans 1 in every register, and doubling the prior scale moves NL CoT's posterior τ by 0.001. Under a deliberately tight prior NL CoT's interval is $[0.99, 2.50]$ — still spanning 1, but by so little that we rely on the conclusion and not the endpoints. Two absolute scales differ here by a factor of two. Marginally, over all 21,328 re-scored cells, Direct succeeds on 8.4% and NL CoT on 10.6% (+2.2pp). The posterior risk difference quoted elsewhere — +0.87pp for NL CoT, +0.24–+0.59pp for the others — is evaluated at the *median* problem for a median model, where the fitted probability is near 1.5%. On neither scale is the effect large, and odds ratios overstate both. It is also not free: NL CoT emits 6.4× Direct's generated characters per verified success (11,256 against 1,754; Appendix O). Two cautions attach to the marginal figure: the five arms do not share a common row set (4,150 *Direct* rows against 3,957 for NL CoT), so it is a *descriptive* rate difference rather than the paired quantity the odds ratios estimate; and the panel is unbalanced, two DeepSeek endpoints contributing n=23 and n=66 five-arm problems (Appendix B) — which is why dropping the DeepSeek family is the most consequential leave-one-out refit. The same yardstick cuts both ways: the same ±2pp reference region used to assess reasoning-label moderation in §7 also contains NL CoT's own risk difference, at posterior probability 0.95. The effect is credible and small, and both halves matter.

What the three levels support. NL CoT alone survives all pooling levels and all 14 common-slope refits. Formal-register harm in the old panel was a scoring artifact (Table 1); their corrected intervals leave average harm unresolved. Every new-model interval spans 1, so these panel averages cannot prescribe a register for an unseen model.

Table 2: **Primary analysis: register effects on proof success vs. *Direct***, 11 models, re-scored panel (§3). Average effects are posterior odds ratios from the pre-specified Bayesian random-slopes fit; the prediction interval is for a new, untested model and holds only under that fit's exchangeability assumption. NL CoT is the only register whose average-effect interval excludes 1, and *every* new-model interval spans 1. Cells truncated at the response-storage cap keep their original verdicts (§3); missingness sensitivity is in §8. Common-slope and meta-analytic pooling, per-model estimates and Holm counts: Appendix C.

Reasoning register	**Average effect** OR [95% HDI]	**New-model effect** 95% prediction interval
NL CoT	**1.58 [1.14, 2.10]**	[0.83, 2.83]
Lean pseudocode	1.19 [0.77, 1.69]	[0.49, 2.69]
Math symbolic	1.37 [0.83, 2.02]	[0.45, 4.00]
Tactic plan	1.15 [0.85, 1.52]	[0.65, 1.90]

4.1 Data integrity under dynamic endpoints

Re-running cells months later showed that nominally identical API endpoints had changed how they respond to CoT while their Direct performance stayed comparatively stable, with two consequences.

First, provider-side HTTP failures must not be counted as proof failures: in the original panel $2{,}151$ of $26{,}140$ regime cells had recorded "insufficient balance" responses as compile failures. We re-ran every affected cell under the identical protocol, and any analysis meeting an unrepaired cell now fails; 27 such cells belong to an endpoint retired before we could re-run them, so that model is excluded whole-row.

Second, mixing collection windows can bias one treatment's estimate more than another's. Pre-audit temporal probes motivated this policy: re-running cells that had been clean all along, each paired with its own earlier outcome (same statement, prompt, temperature and verifier; only the date differs), moved DeepSeek-V4-Flash's Direct accuracy $12.7{\to}14.7\%$ (n.s.) while its NL CoT moved $11.5{\to}24.4\%$ ($p{=}0.021$). Those probe cells predate the audit of §3. Because they have not been re-scored, and the `sorry` defect is response-dependent and need not cancel across dates, they motivated within-window comparison but establish neither the direction nor the magnitude of post-correction drift. When an endpoint's CoT-following improves more than its unprompted accuracy, any panel pooling dates acquires a spurious register effect, and fixing the model name does not make observations exchangeable, because register sensitivity belongs to a model's current state rather than to its identifier. The panel therefore holds every model to one window; contamination counts, the paired drift table and the freeze manifest are in Appendix C.

5 Mechanism Measurement: Teacher-Forced Gold-Proof Interference

If register effects are real but model-dependent, the first candidate explanation is distributional: context in one register may change the probability a model assigns to a correct proof, by a model-dependent amount. We measure that directly, holding the proof fixed. Token-level likelihoods are needed, so this runs on six open-weight models *disjoint from the eleven API endpoints of §4*: it is an independent distributional probe, not a mechanism established for the panel.

Design. On a 200-problem sample (seed-matched to §6) we teacher-force open-weight models on the *identical gold tactic sequence* y^* under four contexts: direct; self-generated NL CoT; self-generated Lean-pseudocode CoT (greedy, with the §4 templates); and the human-written gold NL sketch. We record gold-proof negative log-likelihood (NLL) per token and the per-position KL divergence from the direct context. Unlike API logprobs on self-chosen tokens, this isolates what conditioning does to the probability of a *fixed correct proof*.

Results (three Qwen models, $n{=}200$ each). On Qwen2.5-7B, formal-register context lowers gold-proof likelihood far more than NL does: ΔNLL vs. direct is $+0.038$ nats/token for formal

CoT (93% of problems worsened) against $+0.007$ for NL, an asymmetry of $+0.031$ (paired t-tests, $p<10^{-4}$). On Qwen2.5-Math-7B the two interfere *equally* ($+0.020$ each, $p=0.75$) and on Qwen3-8B the asymmetry is null, so the asymmetry is itself model-dependent. In all three the *gold* NL sketch *raises* gold-proof likelihood (-0.089 to -0.153, $p<10^{-18}$): informative NL content helps, while register-mismatched renderings of the model's own reasoning hurt. That sketch differs in content as well as register, so it cannot separate the two; the transplant of §6 can (Appendix H).

Register distance tracks interference in sample, but does not transfer. Because the asymmetry varies by model, we looked for a model property that orders it. A model's *register distance* is $r_m = \mathrm{NLL}_m(\text{Lean rendering}) - \mathrm{NLL}_m(\text{NL sketch})$ on the same proof content, so larger positive r_m means the NL rendering is the more fluent one for that model. Across six models from four families (adding Mistral-7B-Instruct, Llama-3.1-8B-Instruct and DeepSeek-Prover-V2-7B) it ranges from $+0.09$ (Qwen3-8B) to $+0.45$ (the prover) — the *specialized* prover has the *largest* gap, so specialization alone does not close it — and the register asymmetry orders with r_m (Kendall's rank correlation $= 0.87$, $p=0.02$), with the finer *per-problem* correlation significant in four of the six models and null in two. This supports Hypothesis 2.1(C1) and made register distance a cheap *candidate model-level predictor*, but a pre-registered held-out test rejects that candidacy: on five models from five lineages absent from the design set, frozen on r_m before any interference was measured, the ordering is no better than chance ($4/10$ concordant pairs, exact permutation $p=0.76$), with the extremes reversed. The interference is real and the in-sample association stands, but register distance is a measured regularity, not a predictor (Appendix H).

6 The Transplant Experiment: Register vs. Content

Teacher forcing cannot separate a CoT's register from its content: self-generated NL and formal traces differ in both. We therefore transplant reasoning between registers.

Pilot design and its confound. The model authors an NL and a formal CoT for each problem, *the same model* rewrites each into the opposite register, and proofs are generated under the authorship×presentation cells plus a direct baseline. Our pilot 2×2 on six models had a structural confound: only the *crossed* cells passed through a rewrite, so "matched beats mismatched" conflated register consistency with translation damage.

Equal-rewrite design. We reran it with the number of rewrites *equalized*: seven conditions on three models (Haiku 4.5, DeepSeek-Chat, Qwen-Plus; $n=200$) — direct; NL and formal *raw* (0 rewrites); NL→NL and formal→formal *paraphrases*; NL→formal and formal→NL *translations* (1 rewrite each). All four rewrite prompts instantiate one template differing only in target register, CoT length enters as a covariate, and an independent judge scores semantic fidelity 4.7–$5.0/5$ in every cell, so the rewrites preserve content (Appendix D).

The rewrite, not the register, carries the pilot's effect. Because the models share the same 200 problems, we fit a hierarchical logistic model (presentation $\times$ rewrite class, with $\log$ CoT length and judged fidelity as covariates and random intercepts for problem and model), with cluster bootstraps (5,000 reps). Three planned contrasts locate the pilot's finding in the rewrite rather than in the register. *Rewriting itself is cheap*: paraphrase vs. raw costs -1 proof $[-12, +10]$ ($p=0.92$). *Crossing registers is what costs*: translate vs. paraphrase is -19 $[-33, -5]$ ($p=0.008$), GLMM OR 0.57 $[0.43, 0.77]$ ($p=2\times10^{-4}$). *The presentation-register effect does not survive this control*: among equal-rewrite cells NL presentation leads formal by $+13$ $[-3, +29]$ ($p=0.12$), and the GLMM NL-presented odds ratio is 1.14 $[0.91, 1.43]$ ($p=0.25$) at fixed length and fidelity. A secondary specification entering CoT *authorship* as its own main effect does reach significance for presentation (OR 1.30 $[1.03, 1.63]$, $p=0.02$) with authorship null (OR 0.91, $p=0.40$); we report the planned specification as primary and claim no presentation effect.

The pilot's headline author×presentation "consistency" interaction is larger on re-scored data (OR 2.40 $[1.78, 3.23]$) but still fails the control: it disappears once rewrite steps are equalized, so we read it as translation damage and retract register consistency as causal. An independent renderer likewise does not establish a pooled presentation effect (OR 1.02 $[0.86, 1.21]$; Appendix D). We therefore *withdraw* the presentation effect. What this experiment establishes is narrower: within the equal-

rewrite design, *translating* reasoning from one register into another costs proofs in either direction. We do not claim that cost transfers either: the independent-renderer cells give 144 paraphrase against 148 translation proofs, +4 the other way, though that replication was built for the presentation contrast and is not a powered test of this one (Appendix D).

The per-model cells do not line up behind either contrast: Haiku's raw cells, for instance, are flat (NL 38 vs. formal 39).

7 Do Observable Moderators Explain the Heterogeneity?

Neither mechanism holds beyond its design, which leaves the per-model heterogeneity unexplained. A simpler explanation would be a moderator readable off the model card, and the obvious one is reasoning training: models that generate internal chain-of-thought via RL might respond differently to external CoT (Hypothesis 2.2). We test this across models, as an explicit moderator, and within two model families.

The provider's reasoning label. Four of the eleven endpoints carry a provider-facing reasoning label — MiniMax-M1 (MiniMax Team, 2025), DeepSeek-V4-Pro, o3-mini and Qwen3-30B-A3B-thinking — meaning the provider describes an internal reasoning phase before answering. We test that label, not a verified common training intervention. DeepSeek-R1, whose own report motivates the hypothesis (DeepSeek-AI, 2025), cannot enter the test, because its cells cannot be re-scored (§3); this weakens the test in the direction that matters, since R1 had the most extreme register response we collected.

Label moderation is unresolved under wide equivalence bounds. We fit the label as a shift on the register slope means. Every coefficient straddles zero (γ 95% HDIs: NL CoT $+0.11$ $[-0.41, +0.62]$; Lean pseudocode $+0.34$ $[-0.24, +0.93]$; symbolic math $+0.07$ $[-0.60, +0.78]$; tactic plans $+0.00$ $[-0.46, +0.46]$). A null coefficient cannot by itself establish absence, so we also compute the implied success-probability shift: within $\pm$2pp at posterior probability 0.93–1.00 across the four registers, the largest posterior-mean shift being 0.74pp (Lean pseudocode). With four labeled models these bounds are wide, and effects outside them remain compatible with the data; per-model cells are in Appendix E.

Within-family pairs. Cross-model comparisons confound post-training with base model, scale and data; within-family pairs reduce that without isolating it. Two Qwen3-30B-A3B variants sharing one 30B MoE base (3B active), `-instruct-2507` and `-thinking-2507` at n=500 each, are indistinguishable arm by arm (paired McNemar, all p>0.08), and neither variant's NL-CoT effect separates from the other's (+1.7 vs. +2.7pp; Table S20) — one pair, not a general null. The second pair, DeepSeek-V4-Pro and V4-Flash, would have been the sharpest, but date-matching leaves n=141 paired problems and nothing survives Holm. The full grid is in Appendix M.

These data do not resolve whether the label explains that heterogeneity. Our working hypothesis remains *distributional* (training mixture, register sensitivity at proof-token positions, tuning style), which is what the teacher-forced measurement of §5 begins to quantify.

The task. The other candidate is the task: the panel uses one benchmark, so its register ordering could belong to LeanWorkbook rather than the models. We re-collected miniF2F-test and LeanWorkbook *inside a single window*, on the same problem draws and under the identical protocol and verifier, so the benchmark contrast carries no drift term (§4.1). Only two endpoints survive re-scoring, and on miniF2F both sit at the floor: *Direct* solves 0 of 107 and 4 of 107 regime problems. The power gate fails — the median cell has four *discordant pairs* (problems on which the two conditions disagree, the only ones a paired test uses) and 7 of 8 cells have fewer than ten — so no null here is evidence of absence, and three of the eight difference-in-differences intervals exclude zero, with six changing sign. This comparison is *too imprecisely estimated to support a transfer claim*: raising power, not buying more problems, is what it needs (Appendix L). Two locally run open models separately show a formal-register penalty at 7B scale, including the Lean-specialized DeepSeek-Prover-V2-7B (Ren et al., 2025); template and token budget are confounded there, so we report it as a boundary case and not a claim (Appendix I). That run also fell outside the audit's

re-scoring scope; re-verified for `sorry` only, it stays a boundary case for the prover on both benchmarks and for the generic model on LeanWorkbook only, its miniF2F cells falling to 0–1 authored proof.

8 LIMITATIONS

Scoring and missingness. Every verdict comes from a pipeline we found defective, and 1,205 cells (5.7%) were truncated at a storage cap before the repair and cannot be re-scored. Truncation is register-dependent (0.39% of *Direct* rows against 9.5% of symbolic math), so missingness is not ignorable; we bound it. Dropping them, or imputing at each arm's recovery rate, *raises* NL CoT's estimate (OR 1.69 and 1.65 under the primary model); the adversarial assignment — every unknown *Direct* cell a success, every unknown CoT cell a failure — lowers it to 1.38 $[0.98, 1.83]$. NL CoT's point estimate stays positive in every scenario evaluated, and its interval excludes 1 under all but the adversarial one; tactic plans' common-slope significance does not. The other 857 un-re-scorable cells are determinate: a response with no extractable tactics never reaches the compiler, so failure is its verdict. The runner stored no finish reasons or token usage, so response length cannot show whether a provider hit its limit.

Design and scope. The panel is one benchmark, one prompt family per register, pass@1 — on the two models sampled five times, no register's gain over Direct excludes zero at k=5 (Appendix Q) — and unbalanced: two endpoints contribute n=23 and n=66 five-arm problems. The main panel has no prover baseline, proof search or compiler-feedback loop (a small pilot is in Appendix N), so nothing here says what to deploy — only what a register does with the rest held fixed. Two endpoints are excluded on recoverability, not outcome — one model with the most extreme register response, weakening §7 in the direction that matters. The teacher-forced probe (§5) runs on models *disjoint* from the panel, so it measures distributions, not the panel's mechanism; the second-benchmark comparison is underpowered, not null. Every estimate is indexed to its collection window (§4.1); see Appendix Q.

9 DISCUSSION AND CONCLUSION

What the evidence supports. NL CoT helps on average here, at every pooling level and every leave-one-out refit, but varies by model, and pre-audit probes suggest by collection window; the labeled models do not resolve whether reasoning training moderates it; pre-registered replication confirms neither mechanism. **Distributional conditioning** is one compatible explanation (Chandra et al., 2025; Nikankin et al., 2025), not one the disjoint probe establishes (§5).

Conclusion. A scoring audit changed the answer: on identical data, three significant formal-register harms become none (Table 1). What survives is a small within-panel NL benefit whose new-model interval spans no effect — not a deployment default for an untested endpoint. A register effect is only as trustworthy as the instrument that scored it and the model behind it.

REPRODUCIBILITY STATEMENT

Every proof-success outcome here depends on a Lean 4 compiler verdict, and §3 reports that the pipeline producing those verdicts was itself defective. Reproducibility therefore requires auditing the verdicts as well as the statistics. The anonymous supplement ships *row-level* outcomes — one record per (model, register, problem) cell with the original verdict, the corrected verdict, the sensitivity variant, and why a cell changed or could not be re-scored — plus the repaired verifier and extractor, the five register templates verbatim, the scripts that rebuild every table, a hashed manifest of every frozen artifact, and the claim-checking script we ran against our own manuscript. Two limits are intrinsic: 1,205 truncated responses have unknown corrected verdicts (bounded in §8), while 857 no-tactic responses are determinate failures; API endpoints are not frozen artifacts, so we record identifier, date, temperature and seed per cell. The full apparatus — endpoints, decoding parameters, Lean toolchain (de Moura & Ullrich, 2021) and Mathlib revision (The mathlib Community, 2020) — is in Appendix R.

REFERENCES

Alexander Beiser, David Penz, and Nysret Musliu. Intermediate languages matter: Formal languages and LLMs affect neurosymbolic reasoning. *arXiv preprint arXiv:2509.04083*, 2025.

Abhranil Chandra, Ayush Agrawal, Arian Hosseini, Sebastian Fischmeister, Rishabh Agarwal, Navin Goyal, and Aaron Courville. Shape of thought: When distribution matters more than correctness in reasoning tasks. *arXiv preprint arXiv:2512.22255*, 2025.

Luoxin Chen, Jinming Gu, Liankai Huang, et al. Seed-Prover: Deep and broad reasoning for automated theorem proving. *arXiv preprint arXiv:2507.23726*, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. *International Conference on Automated Deduction (CADE)*, pp. 625–635, 2021.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. arXiv:2501.12948.

Rebecca DerSimonian and Nan Laird. Meta-analysis in clinical trials. *Controlled Clinical Trials*, 7 (3):177–188, 1986.

Joachim Hartung and Guido Knapp. On tests of the overall treatment effect in meta-analysis with normally distributed responses. *Statistics in Medicine*, 20, 2001.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.

Thomas Hubert et al. Olympiad-level formal mathematical reasoning with reinforcement learning. *Nature*, 2025. doi: 10.1038/s41586-025-09833-y.

Albert Q Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. *International Conference on Learning Representations*, 2023. doi: 10.48550/arxiv.2210.12283.

Kimi Team. Kimi K2: Open agentic intelligence, 2025. arXiv:2507.20534.

Ryan Liu, Jiayi Geng, Addison J Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L Griffiths. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse. *arXiv preprint arXiv:2410.21333*, 2024.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12:153–157, 1947.

MiniMax Team. MiniMax-M1: Scaling test-time compute efficiently with lightning attention, 2025. arXiv:2506.13585.

Yaniv Nikankin, Anja Reusch, Aaron Mueller, and Yonatan Belinkov. Arithmetic without algorithms: Language models solve math with a bag of heuristics. *International Conference on Learning Representations*, 2025.

Z. Z. Ren, Zhihong Shao, Junxiao Song, Huajian Xin, et al. DeepSeek-Prover-V2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801*, 2025.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design. *International Conference on Learning Representations*, 2024. doi: 10.48550/arxiv.2310.11324.

Parshin Shojaee et al. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *arXiv preprint arXiv:2506.06941*, 2025.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To CoT or not to CoT? chain-of-thought helps mainly on math and symbolic reasoning. *International Conference on Learning Representations*, 2025. doi: 10.48550/arxiv.2409.12183.

The mathlib Community. The Lean mathematical library. *International Conference on Certified Programs and Proofs (CPP)*, 2020.

Sumanth Varambally, Thomas Voice, Yanchao Sun, Zhifeng Chen, Rose Yu, and Ke Ye. Hilbert: Recursively building formal proofs with informal reasoning. *NeurIPS MATH-AI Workshop*, 2025.

Yue Wang et al. Thoughts are all over the place: On the underthinking of o1-like LLMs. *arXiv preprint arXiv:2501.18585*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 2022.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22:209–212, 1927.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. *International Conference on Learning Representations*, 2022.

Kaiyu Yang, Aidan Swope, Alex Gu, Raghuraman Chalapathy, Peiyang Song, Shixing Peng, Shuze Yu, and Steven Lemieux. LeanDojo: Theorem proving with retrieval-augmented language models. *Advances in Neural Information Processing Systems*, 2023. doi: 10.48550/arxiv.2306.15626.

Huaiyuan Ying et al. Lean Workbook: A large-scale Lean problem set formalized from natural language math problems. *Advances in Neural Information Processing Systems*, 2024.

Chengshuai Zhao, Zhen Tan, et al. Is chain-of-thought reasoning of LLMs a mirage? a data distribution lens. *arXiv preprint arXiv:2508.01191*, 2025.

USE OF GENERATIVE AI

Generative AI tools were used to assist with code development and debugging, data-analysis scripting, figure preparation, and language editing. All research questions, experimental design, statistical decisions, result interpretation, and scientific claims were determined and verified by the authors. All reported numerical values were regenerated from the archived experimental data by the frozen analysis pipeline released with the paper and independently checked by the authors against that pipeline's output; all citations were author-verified against the cited works.

A PROMPT CONDITIONS

The independent variable is the instruction and reasoning slot preceding a shared proof slot. All five conditions begin "You are a Lean 4 theorem prover" and end with the target theorem, '`:= by`', and a condition-specific slot. They reuse the same ten theorem–tactic demonstrations (including `nlinarith`, `ring`, `omega`, `simp`, `field_simp`, and `norm_num`); only the instruction and reasoning line in each demonstration change. The verbatim generators and all ten demonstrations are in `experiments/epistemic/cot_variants.py` in the anonymous supplement.

Direct. "Given a theorem statement, provide ONLY the proof tactic(s). No explanations." The target ends at `-- Proof:`.

NL CoT. "First explain your reasoning in natural language (1–2 sentences after '– Reasoning:'); then provide the proof tactic(s) after '– Proof:'." The target ends at `-- Reasoning:`.

Lean pseudocode. "First write a brief tactic plan using Lean-style pseudocode (after '– Plan:'). Format: [step1] → [step2] → [tactic]; then provide the proof tactic(s) after '– Proof:'." The target ends at `-- Plan:`.

Symbolic math. "First write a brief mathematical justification using symbols (after '– Math:'). Use ∵ (because), ∴ (therefore), and standard math notation; then provide the proof tactic(s) after '– Proof:'." The target ends at `-- Math:`.

Tactic plan. "First list 1–3 candidate Lean 4 tactics to try, ranked by likelihood (after '– Tactics to try:'); then provide the proof tactic(s) after '– Proof:'." The target ends at `-- Tactics to try:`.

These conditions change more than an abstract register: they also change the slot marker, instructions, and response length. The experiment estimates the effects of these *prompt packages*; the paraphrase check in Appendix Q probes wording for two registers on a limited grid; surface prompt form is known to move model behaviour on its own (Sclar et al., 2024), and CoT's measured benefit concentrates on math and symbolic tasks (Sprague et al., 2025). It does not isolate a pure register effect across all models.

B Success-Rate Definitions and Data Audits

Per-model panel counts. Table S1 gives the LLM-contributed proof counts behind every pooled estimate, per model and register, generated from the frozen primary bundle. The n column is the number of regime problems present in *all five* arms for that model after the era policy of §4.1. It falls into three groups, which is the cost of date-matching made explicit: **six** endpoints have the complete five-arm grid at n=402 (MiniMax-M1, Haiku, DeepSeek-chat, GLM-4-Plus, Qwen-Plus, Qwen3-instruct); **two** lose a single cell to the policy and sit at n=401 (GPT-4o-mini in NL CoT, o3-mini in Lean pseudocode, one re-run cell each dropped as off-date); and **three** lose substantially more (Qwen3-thinking 367, DeepSeek-V4-Flash 66, DeepSeek-V4-Pro 23), because large parts of their grids were re-collected in the later window and the primary policy will not mix windows within a model. Because n is a five-arm intersection it is smaller than the two-arm pairings quoted for individual contrasts in the main text, and smaller than the row set the regressions use. The pooled fits take every regime row an arm has, not the five-arm intersection: 4,150 *Direct* rows, 3,957 NL CoT, 4,404 Lean pseudocode, 4,404 symbolic math and 4,413 tactic plans, which is the 21,328 total quoted throughout, carrying 2,256 successes. This table is therefore the authoritative source for *per-model* counts on a common problem set, not for the regressions' denominators; a per-arm breakdown is in the frozen bundle.

Table S1: Per-model LLM-contributed proofs by register, primary (era-matched) analysis, 11 models. n = regime problems present in all five arms. Generated by `paper-iclr/tables/build_frozen_tables.py` from the frozen bundle.

Model	n	Direct	NL	Psd.	Sym.	Tac.
MiniMax-M1	402	29	27	22	17	19
claude-haiku-4-5-20251001	402	85	99	109	116	105
deepseek-chat	402	45	75	71	67	54
deepseek-v4-flash	66	10	18	15	18	14
deepseek-v4-pro	23	5	7	6	6	7
glm-4-plus	402	12	23	17	29	17
gpt-4o-mini	401	4	7	4	3	3
o3-mini	401	13	11	14	12	15
qwen-plus	402	23	24	13	29	20
qwen3-30b-a3b-instruct-2507	402	17	24	17	14	21
qwen3-30b-a3b-thinking-2507	367	16	26	18	12	11

Three success-rate definitions. The main paper reports the LLM-contribution regime (LLM successes over the 402 problems the 16-tactic baseline leaves unsolved). The other two conventions — LLM-only over all 500, and hybrid (simple tactics + LLM) over all 500 — are in `audit_accuracy_defs.csv` in the code appendix, so any cell can be re-derived under a different denominator. Note that file is generated by a script that reads the raw result JSONs and therefore predates the era policy and the whole-row exclusion; the counts in Table S1 are the authoritative ones.

Provenance of the 764 recoveries. The correction's direction rests on the recovered successes, so they are audited exhaustively rather than by sample. For every one of the 764 cells whose verdict moved from failure to success, we re-ran the repaired extractor against the model's own stored response and asked whether each extracted tactic line occurs *verbatim* in that response. It does in 764 of 764 — Direct 36, NL CoT 101, Lean pseudocode 232, symbolic math 220, tactic plans 175 — with no cell yielding an empty extraction and none missing its response text (`scripts/recovery_audit.py`, `recovery_audit.json`). This bounds one channel and not another: it establishes that the repair never fabricated proof text, but it cannot establish that a compilable block lifted from a model's Lean-like CoT is what that model *offered* as its proof, an opportunity the formal registers have and Direct structurally does not.

Response-length, truncation, and extraction audits. Experiment JSONs store each response truncated at a 4,000-character cap, and record the untruncated length alongside it; that cap is why 1,205 cells cannot be re-scored (§3). Re-running the tactic extractor over every stored response: the *empty*-extraction rate is $\leq 5.5\%$ in the worst cell (GLM-4-Plus, math symbolic). That statistic counts only cells where extraction returned nothing, which is why it read as safe: the defect this paper reports is non-empty *wrong* extraction, invisible to that count and present in 30.7% of cells. Kimi K2-Thinking is a special case, and Table S2 separates its failure modes, because the exclusion of its row should be auditable rather than taken on trust. Per-cell response-length distributions, empty/API-error counts, retry counts, and timing are in `audit_lengths_parsing.csv` in the code appendix. Failure families for three models, computed on the pre-correction failure set and reported as provenance only, are in Appendix F.

Table S2: Failure-mode audit of the excluded Kimi-K2-Thinking row, per condition (500 problems each). "Empty content" cells returned HTTP 200 with reasoning tokens but no visible answer content. They would count as failures under the stated scoring rule; they are not grounds for exclusion. The exclusion is driven by permanently missing provider-error cells and the inability to reconstruct a contemporaneous complete row. No corrected Kimi-inclusive pooled fit is available.

Condition	API failure	Empty content	Extraction failure
Direct	19	121	0
NL CoT	6	145	1
Lean pseudocode	2	131	2
Math symbolic	0	139	3
Tactic plan	0	138	2
Total	**27**	674	8

The 27 API failures are the unrepairable cells: they are missing observations, never compile failures, and the endpoint became unavailable before they could be re-run. The empty-content cells are different — valid HTTP responses that yield no proof and would score as failures under the stated rule. We do not use Kimi's older stored verdicts as a corrected-data sensitivity fit: its proof extraction was not re-scored under the repaired pipeline. The exclusion criterion is provenance: 27 cells are permanently missing, concentrated in Direct and NL CoT (the two arms of the headline contrast), and the endpoint was retired before they could be re-run, so no contemporaneous complete row can be reconstructed for this model under any era policy.

C Pooling Details and Sensitivity

The main text reports the primary fit only (Table 2). The full three-level pooling table is Table S3 and the per-model estimates behind it are Figure S1; both are reproduced here unchanged.

Estimator Sensitivity of the Main Panel. Appendix L shows the second-benchmark interaction is interval-sensitive to the estimator. The same question put to the main panel gives the opposite answer, and we report it because it is the first thing that check invites. A cluster-robust marginal logit on all 21,328 re-scored regime rows (clustering on problem, models as fixed effects, no random effects at all) returns NL CoT OR 1.41 $[1.25, 1.59]$; Lean pseudocode 1.30 $[1.15, 1.47]$; symbolic math 1.37 $[1.22, 1.54]$; and tactic plans 1.12 $[0.99, 1.26]$, p=0.06 — the one register whose common-

Table S3: **Register effects on proof success at three levels of pooling** vs. *Direct*, 11 models, re-scored panel (§3). Every register's point estimate is above 1; NL CoT is the only one whose primary-model interval excludes 1, and every new-model prediction interval spans 1. Definitions and full fits: Appendix C.

Variant	(i) Common slope OR [95% CI]	(ii) Meta-analysis RE OR / HK CI / I^2	(iii) Random slopes (primary) OR [95% HDI] / $P(<1)$ / new-model PI
NL CoT	**1.691 [1.482, 1.930]***	**2.05 / [1.25, 3.36] / 48%**	**1.58 [1.14, 2.10]** / .003 / [0.83, 2.83]
Lean pseudocode	1.502 [1.329, 1.697]*	1.36 / [0.86, 2.17] / 71%	1.19 [0.77, 1.69] / .18 / [0.49, 2.69]
Math symbolic	1.629 [1.444, 1.838]*	1.44 / [0.78, 2.66] / 76%	1.37 [0.83, 2.02] / .07 / [0.45, 4.00]
Tactic plan	1.179 [1.038, 1.340]*	1.22 / [0.82, 1.81] / 45%	1.15 [0.85, 1.52] / .15 / [0.65, 1.90]

Per-model McNemar after Holm across 44 cells (sig. harm / sig. benefit; directional in parentheses):
NL CoT 0 (2) / 1 (9) Lean pseudocode 0 (3) / 2 (6), 2 ties Tactic plan 0 (4) / 0 (7) Math symbolic 0 (5) / 2 (6)

*Holm-significant at α=0.05 (4 non-reference variants).

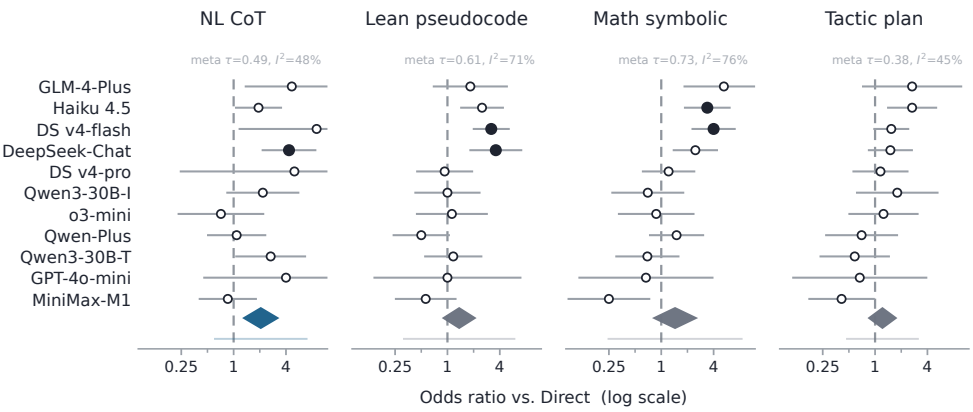


Figure S1: **Register effects are heterogeneous; only NL CoT shows a robust positive panel-average effect.** Points and whiskers: per-model conditional odds ratios vs. Direct with 95% CIs, 11 models ordered by mean formal effect, primary analysis; filled markers are Holm-significant McNemar cells (44-cell panel). Diamonds: random-effects pooled estimates with Hartung–Knapp intervals, as tabulated; the thin bar beneath each is the 95% prediction interval for a new model. These, and the annotated τ/I^2, are *meta-analytic* — a different estimand from the random-slopes τ=0.17–0.47 and prediction intervals in the text. Every panel still contains models on both sides of 1.

slope significance does not survive dropping the random effects. Every direction matches Table S3. The odds ratios are attenuated toward 1 because they are marginal rather than conditional. The panel carries roughly three times the rows of the interaction subset across four times as many models, which is why variational variance underestimation matters for two of four interaction fits. These four fits are the baseline row of `lomo_cluster_robust.csv`.

Two-stage meta-analysis. Table S4 gives the per-register random-effects estimates: τ^2 by DerSimonian–Laird (DerSimonian & Laird, 1986), intervals and p-values by the Hartung–Knapp (Hartung & Knapp, 2001) small-sample adjustment (t-based, with dispersion-re-estimated SE). We report HK rather than the uncorrected normal interval because k=11 is small enough for the normal interval to over-reject, and we use one interval method throughout so that the same estimate never appears with two different intervals: the uncorrected intervals are narrower ($[1.32, 3.18]$ for NL CoT against HK's $[1.25, 3.36]$) and would not change any verdict. NL CoT survives HK (p=0.009) while all three formal registers remain non-significant, so the main text's reading — NL is the only register that survives random-effects pooling, and the formal registers lose significance to between-model heterogeneity — is conservative-robust.

Leave-one-out refits. The common-slope GLMM is refit 14 times: once per model dropped (11) and once per multi-model family dropped. Three providers contribute more than one endpoint —

Table S4: Random-effects meta-analysis per register (11 models, primary era-matched analysis; conditional log-ORs from paired discordant counts). τ^2 by DerSimonian–Laird; intervals and p-values by the Hartung–Knapp small-sample adjustment, the same interval method quoted in Table S3. Generated by `meta_heterogeneity.py`; emitted from the frozen bundle by `build_frozen_tables.py`.

Variant	OR	95% CI (HK)	p (HK)	τ	I^2
NL CoT	2.05	[1.25, 3.36]	0.009	0.49	48%
Lean pseudocode	1.36	[0.86, 2.17]	0.164	0.61	71%
Math symbolic	1.44	[0.78, 2.66]	0.212	0.73	76%
Tactic plan	1.22	[0.82, 1.81]	0.298	0.38	45%

DeepSeek (3), Qwen (3), OpenAI (2), covering 8 of the 11 models — so the remaining *three* endpoints (Anthropic, Zhipu, MiniMax) are singletons, for which dropping the family is identical to dropping the model and is not counted twice. Counting the full-panel fit alongside the refits gives the 15 fits recorded in `lomo_results.csv`. Per register, the number of refits in which the effect stays significant at α=0.05 is: NL CoT $11/11$ models and $3/3$ families; symbolic math $11/11$ and $3/3$; Lean pseudocode $11/11$ and $2/3$; tactic plans $8/11$ and $2/3$. Every fit that breaks a register is a DeepSeek one: dropping the DeepSeek family is the single refit that unseats Lean pseudocode, and three of the four tactic-plan failures drop a DeepSeek endpoint or the family. The DeepSeek models supply the largest share of re-scored cells, which is the sense in which the corrected panel's weaker registers rest on them. The CSV's baseline row reports the full-panel fit and matches Table S3 coefficient for coefficient.

Cluster-robust replication of the leave-one-out refits. Those 14 refits use the same variational-Bayes backend as the main fit, and Appendix L shows that backend understates variance on a small interaction fit, so a reader is entitled to ask whether the leave-one-out counts inherit the problem. We therefore repeated all 14 refits with the cluster-robust marginal logit above — GEE, exchangeable working correlation, clustering on problem, models as fixed effects, sandwich standard errors — which makes no random-effects assumption at all. **NL CoT is significant in $11/11$ model refits and $3/3$ family refits under *both* estimators**, with OR between 1.25 and 1.47 and no sign change in any refit, so the headline robustness count does not rest on the variational backend. Symbolic math and Lean pseudocode also reproduce ($11/11$ and $3/3$; $11/11$ and $2/3$), which the pre-correction data did not. Tactic plans are the one register the estimator moves: $4/11$ models and $1/3$ families under the marginal fit, against $8/11$ and $2/3$ under the variational one. No refit changes sign under either estimator. We therefore treat the *tactic-plan* leave-one-out count as estimator-dependent and do not offer it as robustness evidence, which is why the claims hierarchy of §4 grades tactic plans lowest. Per-refit estimates in `lomo_cluster_robust_v4.csv`.

Bayesian hierarchical model. The primary analysis is a Bayesian hierarchical logistic regression fit with numpyro's NUTS. Specification: $\text{logit}\, p_{mvi} = \alpha + \beta_v + u_m + w_{mv} + q_i$ with

$$\alpha \sim \mathcal{N}(0, 2.5), \quad \beta_v \sim \mathcal{N}(0, 1), \quad u_m \sim \mathcal{N}(0, \sigma_m), \quad q_i \sim \mathcal{N}(0, \sigma_q), \quad w_{mv} \sim \mathcal{N}(0, \tau_v),$$

$\sigma_m \sim \mathcal{N}^+(0, 1)$, $\sigma_q \sim \mathcal{N}^+(0, 1.5)$, $\tau_v \sim \mathcal{N}^+(0, 1)$, all on the log-odds scale. The random slopes are non-centered ($w_{mv} = \tau_v z_{mv}$, $z_{mv} \sim \mathcal{N}(0, 1)$) and independent across registers rather than sharing a full correlation matrix, which 11 model-level groups cannot identify. Because τ_v is estimated from those 11 groups it is prior-sensitive by construction, and its posteriors place substantial mass near zero (Table S5), so the new-model prediction intervals are prior-regularized and should not be read as tight frequentist coverage. Four chains, 1,000 warmup and 1,000 retained draws each; $\hat{R} \leq 1.01$ and minimum bulk effective sample size 449 across all reported quantities. The frozen run's sampler diagnostics were not written to disk at the time, so we recovered them by refitting under the identical model, data, sampler configuration and seed; that refit reproduces the frozen estimates exactly (NL CoT OR 1.583, $\hat{R}$ 1.01, min bulk ESS 449), which is what licenses reporting its diagnostics for the frozen fit: 0 divergent transitions in 4,000 post-warmup draws, no iteration at the maximum tree depth (mean depth 6.5), per-chain BFMI 0.66–0.75, and minimum tail ESS 844 (`scripts/nuts_diagnostics.py`). Table S5 reports posterior summaries including the posterior predictive interval for an untested model; Table S6 converts them to absolute risk differences at the panel base rate, which is the scale the odds ratios overstate.

The reasoning-label equivalence test is computed on that absolute scale: for each register we draw the implied success-probability difference between a labeled-reasoning and a labeled-standard model at the panel-average base rate, and report the posterior probability that its magnitude is below 2 percentage points (the ROPE). Those probabilities are 0.93–1.00 (largest 0.997, for tactic plans); the corresponding interaction coefficients on the log-odds scale all straddle zero.

Table S5: Bayesian hierarchical logistic regression with per-model random slopes, primary (era-matched) analysis, 11 models. OR: panel-average odds ratio vs. Direct with 95% HDI; τ: between-model SD of the log-OR; PI: 95% prediction interval for a new model's OR; $P(\text{harm})$: posterior probability a new model is harmed. Emitted from the frozen bundle by `build_frozen_tables.py`.

Variant	OR [95% HDI]	$P(\text{OR}<1)$	τ	New-model PI	$P(\text{harm})$
NL CoT	1.58 [1.14, 2.10]	0.00	0.21	[0.83, 2.83]	0.06
Lean pseudocode	1.19 [0.77, 1.69]	0.18	0.37	[0.49, 2.69]	0.32
Math symbolic	1.37 [0.83, 2.02]	0.07	0.47	[0.45, 4.00]	0.26
Tactic plan	1.15 [0.85, 1.52]	0.15	0.17	[0.65, 1.90]	0.23

Table S6: Absolute risk differences implied by the primary random-slopes posterior, at the LLM-regime base rate. $P(|\Delta\text{RD}|<2\text{pp})$ is the ROPE mass for the *register effect itself* against Direct, reported as an absolute-scale description of how small these effects are in percentage points. It is *not* the reasoning-label equivalence quantity, which is a different posterior on a different contrast and is given separately in Table S7. Emitted from the frozen bundle by `build_frozen_tables.py`.

Variant	RD (pp)	95% HDI	$P(\lvert\Delta\text{RD}\rvert<2\text{pp})$
NL CoT	+0.87	[+0.10, +2.06]	0.95
Lean pseudocode	+0.30	[-0.45, +1.10]	0.99
Math symbolic	+0.59	[-0.27, +1.82]	0.97
Tactic plan	+0.24	[-0.23, +0.94]	1.00

Table S7: **The reasoning-label equivalence test** — a different estimand from Table S6. γ is the label's shift on the register slope (log-odds); the implied RD shift is the difference in success probability between a labeled-reasoning and a labeled-standard model at the panel-average base rate; the final column is the posterior probability that this *shift* is smaller than 2pp, which is the quantity quoted as 0.93–1.00 in the main text. Table S6 instead asks whether each register's effect against Direct is small; both are ROPE masses, but of different contrasts, and only this one bears on whether the reasoning label moderates. Emitted from the frozen bundle by `build_frozen_tables.py`.

Variant	γ [95% HDI]	Implied RD shift (pp)	$P(\lvert\Delta\text{RD}_{\text{shift}}\rvert<2\text{pp})$
NL CoT	+0.110 [-0.41, +0.62]	+0.29	0.971
Lean pseudocode	+0.343 [-0.24, +0.93]	+0.74	0.932
Math symbolic	+0.070 [-0.60, +0.78]	+0.18	0.965
Tactic plan	+0.001 [-0.46, +0.46]	+0.01	0.997

D TRANSPLANT: PILOT CELLS AND EQUAL-REWRITE CELLS

Pilot 2×2 (six models). Table S8 gives the pilot's per-model cells. Only the crossed cells pass through a rewrite in this design, which is the confound the equal-rewrite redesign removes; the pilot's author×presentation interaction did not survive that control and is retracted in the main paper.

Equal-rewrite cells (three models). Table S9 repeats the equal-rewrite per-model cells; the GLMM summary, planned-contrast bootstrap distributions, and per-pair fidelity judgments are in `transplant_v2_glmm_summary.txt`, `transplant_v2_contrasts.csv`, and `fidelity_judgments_*.json`.

Table S8: *Pilot* transplant 2×2 under the same-model translation pipeline, re-scored (n=157 regime problems; LLM-contributed proofs). Pres. Δ = (NL→NL + F→NL) − (NL→F + F→F); Auth. Δ analogous.

Model	Direct	NL→NL	NL→F	F→NL	F→F	Pres. Δ	Auth. Δ
DeepSeek-Chat	22	26	15	17	13	+15	+11
GLM-4-Plus	6	15	1	7	2	+19	+7
GPT-4o-mini	0	2	0	0	0	+2	+2
Haiku 4.5	28	33	31	34	42	-6	-12
Qwen-Plus	8	10	3	11	5	+13	-3
Qwen3-30B-instr	6	7	2	3	0	+8	+6

Table S9: Equal-rewrite transplant cells, re-scored (n=157 regime problems; LLM-contributed proofs). Generated by `transplant_v2_hierarchical.py`.

Model	Direct	NL raw	F raw	NL para	F para	NL→F	F→NL
claude-haiku-4-5-20251001	36	38	39	35	40	33	37
deepseek-chat	8	14	13	16	14	8	13
qwen-plus	8	10	6	11	3	5	4

Independent-renderer replication (E9). The equal-rewrite design has each subject model rewrite its own CoT, so its renderer is not independent of its subject. E9 closes that gap and nothing else: the same three subject models, the same seed-42 n=200 problems (157 in the LLM-contribution regime), the same authored CoTs read verbatim from the equal-rewrite artifacts, and the same four rewrite prompts — but every rewrite is performed by `gpt-4o-mini`, which shares no lineage with any subject and never generates a proof. The protocol, runner, analysis code, and the PASS/FAIL rule were frozen and hashed before the first API call (`docs/E9_INDEPENDENT_RENDERER_PRESPEC_20260901.md`; snapshot digest `11216d95...`, `e9_outcome_measured: false`). The primary estimand is the presentation main effect; the pre-declared estimator is a cluster-robust GEE marginal logit (exchangeable working correlation, clustering on problem, model fixed effects, log rendering length and independently judged fidelity as covariates). Mean-field VB was excluded by protocol.

The effect does not replicate (Table S10): pooled presentation OR 1.02 $[0.86, 1.21]$, p=0.79, against 1.14 $[0.91, 1.43]$ (p=0.25) under self-rewriting — so after re-scoring neither design shows the effect. The pooled null is not uniform, and we name the exception rather than pool it away: on Qwen-Plus the presentation odds ratio is 2.89 $[1.59, 5.24]$ ($p=5\times10^{-4}$), while Haiku (1.13) and DeepSeek-Chat (0.99) are flat. A NUTS companion on the pooled fit agrees with the null (OR 0.93 $[0.59, 1.55]$). A paired cluster bootstrap over problems gives $+10$ proofs $[-5, +25]$ (p=0.23); the authorship main effect (1.17 $[0.84, 1.63]$) and the authorship×presentation interaction (0.86 $[0.60, 1.24]$) are both null. By the frozen rule this is FAIL rather than a mixed result: a pooled interval containing 1 is failure however the per-model estimates fall.

Three integrity facts. The manipulation held — 92.9% of formal-presented renderings carry the template's bracketed-arrow format against 0% of NL-presented ones — so the presentation factor is real and not merely nominal. All 471 problem×model units completed with zero API errors and zero empty renderings, inside the protocol's 5% INCOMPLETE guard. And an independent judge (MiniMax-M1, independent of both the renderer and every subject) scores content preservation at 4.55–5.00 across all twelve model×operation cells, so the null is not rewrite damage. We note for transparency that `gpt-4o-mini`, the E9 renderer, was the fidelity *judge* in the equal-rewrite experiment; the two roles never coincide within one experiment, which is why E9 judges with a third model rather than reusing it. One wrinkle we report rather than adjust: the renderer reproduced the bracketed format in only 58.6% of DeepSeek's formal-to-formal paraphrases, against $\geq 98.7\%$ in every other formal cell.

We deliberately do *not* fit a renderer×presentation interaction across the two experiments. The *Direct* anchor moved between the July and September windows — DeepSeek-chat $+17$ proofs, Haiku -3, Qwen-Plus $+2$ — so a cross-experiment contrast would confound renderer identity with

the endpoint drift of §4.1. E9's own contrast is immune to that drift because all four of a model's cells were generated in one window from fixed source text. The two intervals also overlap, so E9 shows that the presentation effect does not replicate under an independent renderer, not that it reverses. The single model that does show it, Qwen-Plus, does so on 29 successes spread across four cells, and its adjusted OR is inflated relative to its raw 18 versus 11 split by the length covariate.

Table S10: E9: independent-renderer replication of the equal-rewrite cells (n=157 regime problems per model; LLM-contributed proofs). Columns are authorship→presentation. PΔ = (N→N + F→N) − (F→F + N→F). Per-model ORs drop the fidelity covariate, which is collinear with the 2×2 within one model (pre-data amendment E9-A1). Generated by `e9_analysis.py` + `e9_tables.py`.

Model	Dir	N→N	F→F	N→F	F→N	PΔ	Pres. OR [95% CI]
Haiku 4.5	33	35	37	34	38	+2	1.13 [0.89, 1.43]
DeepSeek-chat	25	30	26	33	30	+1	0.99 [0.78, 1.26]
Qwen-Plus	10	11	5	6	7	+7	2.89 [1.59, 5.24]
Pooled	cluster-robust GEE, model FE, length + fidelity						**1.02** [0.86, 1.21]

E Reasoning Models: Per-Model Results

With DeepSeek-R1 excluded (§3), MiniMax-M1 is the only labeled-reasoning endpoint with a complete five-arm grid; its per-register counts appear in Table S1 alongside every other endpoint, and none of its cells is Holm-significant. R1 was the endpoint this section was originally built around, and its loss is the main reason the moderator test of §7 is underpowered.

The excluded endpoint, descriptively. Kimi K2-Thinking (Kimi Team, 2025) was collected under the same protocol and is descriptively interesting — NL -5.0pp and math-symbolic -4.8pp relative to Direct, nominal McNemar $p<0.003$, with Lean pseudocode and tactic planning neutral — but it is excluded from every estimate in the paper (§4.1), so we do not count it as evidence for or against moderation. Those counts are additionally *pre-audit*: `rescore_v3.py` excludes this endpoint by name, so they were produced by the defective scorer of §3 and are reported as provenance for the exclusion, not as measurements.

F Compiler Error Taxonomy

All stored failed proof attempts for three representative models (Haiku 4.5, DeepSeek-Chat, Qwen-Plus; 5,270 failures) were recompiled locally to capture real compiler diagnostics and classified into failure families (Table S11). Under the *then-current* extractor, zero stored failures flipped to success on recompilation; the 764 recoveries of §3 come from the repaired extractor, which sends the compiler different text. The reversal—Direct fails semantically, formal registers fail syntactically—holds within each model individually (per-model breakdown in `error_taxonomy_refined.csv`). We no longer use that reversal to motivate §5, and it should not be read as a register effect. The extraction defect of §3 prepended restated theorem headers and echoed exemplars to formal-register outputs in 890 failing cells and in none under *Direct*, and a theorem declaration followed by tactics produces a parse error mechanically. Because this taxonomy was computed on the pre-correction failure set, its register×syntax pattern is confounded with that defect; it would have to be recomputed on corrected extractions before it could support any claim, and we report it as provenance only.

Table S11: Failure-family distribution per CoT variant, pooled over Haiku 4.5, DeepSeek-Chat, and Qwen-Plus. *Syntax/parse*: malformed tactics; *Lean 3 naming*: Lean 3 identifier contamination; *Unknown identifier*: hallucinated names/tactics; *Tactic gave up*: legal tactic, goal not closed (e.g., `linarith` failure); *Unsolved goals*: incomplete proof. Generated by `error_taxonomy_table.py`.

Variant	Syntax / parse	Lean 3 naming	Unknown identifier	Tactic gave up	Unsolved goals	Other	n
Direct	7.8%	6.5%	10.1%	66.2%	6.4%	3.0%	1039
NL CoT	32.4%	8.3%	9.6%	35.4%	12.6%	1.7%	1010
Lean pseudo	48.6%	10.0%	11.4%	19.4%	8.9%	1.8%	1085
Math symbolic	44.3%	9.0%	14.5%	19.9%	10.6%	1.7%	1059
Tactic plan	51.0%	10.3%	10.4%	19.9%	7.3%	1.1%	1077

G Generation-Confidence Probe (Exploratory)

The API-level probe recorded mean log-probability of each model's *own generated* proof-phase tokens under direct/NL/formal conditioning (n=200). Because token sets differ across conditions, this is a confidence diagnostic, not a likelihood comparison on fixed content (the teacher-forced measurement in the main paper is the clean version). GPT-4o-mini operates at a different absolute logprob scale and is excluded from interpretation. Table S12 reports the per-model means and paired differences.

Table S12: Mean proof-token log-probability (n=200); higher = more confident. Δ rows are paired differences vs. Direct (paired t-test).

Condition	**Qwen-Plus**	**GPT-4o-mini**	**DS-Chat**	**v4-flash**	**v4-pro**
Direct	-0.081	-0.421	-0.037	-0.093	-0.066
NL CoT	-0.042	-5.056	-0.032	-0.171	-0.038
Formal CoT	-0.054	-6.813	-0.039	-0.082	-0.041
Δ(NL $-$ Dir.)	$+.039^{***}$	-4.64^{*}	$+.005$	$-.078^{**}$	$+.028^{*}$
Δ(Formal $-$ Dir.)	$+.027^{**}$	-6.39^{*}	$-.002$	$+.011$	$+.025^{*}$
Δ(NL $-$ Formal)	$+.012^{*}$	$+1.73$	$+.007$	$-.089^{**}$	$+.003$

$^{***}p<0.001$, $^{**}p<0.01$, $^{*}p<0.05$

Notably, on Qwen-Plus the register that hurts compiled success still *raises* generation confidence—the confidence/success dissociation cited in the main paper. On v4-flash, NL conditioning reduces generation confidence (-0.078, $p<0.01$). We no longer read that as corroborating an observational NL penalty for this endpoint: the apparent penalty was an artifact of miscounted API failures and is withdrawn (§4.1), so the confidence shift stands as an unexplained dissociation rather than as supporting evidence.

H Teacher-Forced Measurement: Full Results

Table S13 gives the full teacher-forced results for all three Qwen models (n=200 each, seed 42). ΔNLL is the paired difference vs. the direct context on the identical gold tactic sequence; KL is mean per-position full-vocabulary divergence from direct; % worse is the fraction of problems with positive ΔNLL. Wilcoxon signed-rank tests agree with all reported t-tests.

Cross-model ordering (six models, four families). The formal-vs-NL interference asymmetry orders with each model's register distance r_m across all six models—Qwen2.5-7B (r_m=0.324, asym $+0.031$), Qwen2.5-Math-7B (0.299, $+0.001$), Qwen3-8B (0.095, -0.002), Mistral-7B-Instruct (0.316, $+0.003$), Llama-3.1-8B-Instruct (0.334, $+0.008$), DeepSeek-Prover-V2-7B (0.447, $+0.075$)—giving Kendall's rank correlation $= 0.87$ (p=0.02, n=6; the single reordering is Llama,

Table S13: Teacher-forced gold-proof interference, all models and contexts. Generated from `gold_nll_summary.csv` (`gold_nll_analysis.py`).

Model	Context	ΔNLL	t-test p	KL	% worse
Qwen2.5-7B	self NL CoT	$+0.0071$	1.1×10^{-5}	0.0152	64.5
	self formal CoT	$+0.0381$	3.9×10^{-37}	0.0366	93.0
	gold NL sketch	-0.1305	5.6×10^{-33}	0.2477	13.1
Qwen2.5-Math-7B	self NL CoT	$+0.0196$	2.7×10^{-18}	0.0307	82.5
	self formal CoT	$+0.0203$	3.4×10^{-19}	0.0355	80.5
	gold NL sketch	-0.0894	1.7×10^{-19}	0.2422	23.1
Qwen3-8B	self NL CoT	$+0.0130$	7.8×10^{-8}	0.0279	66.5
	self formal CoT	$+0.0112$	1.2×10^{-5}	0.0327	64.5
	gold NL sketch	-0.1530	2.4×10^{-36}	0.274	10.1
Mistral-7B-Inst	self NL CoT	-0.0083	0.00273	0.0548	45.5
	self formal CoT	-0.0058	0.0442	0.0433	48.5
	gold NL sketch	-0.1469	1.8×10^{-27}	0.3198	13.6
Llama-3.1-8B	self NL CoT	$+0.0102$	2.1×10^{-6}	0.0354	63.5
	self formal CoT	$+0.0183$	2.5×10^{-13}	0.0361	69.0
	gold NL sketch	-0.1202	1.8×10^{-22}	0.2641	18.1
DeepSeek-Prover-V2-7B	self NL CoT	-0.0718	1.0×10^{-12}	0.2519	24.0
	self formal CoT	$+0.0032$	0.754	0.2483	47.0
	gold NL sketch	-0.1046	1.5×10^{-13}	0.3825	22.1

whose asymmetry is slightly below its r_m rank). The *within-model* per-problem correlation between register distance and formal-CoT interference is significant in four of the six models—strong on Qwen2.5-7B (ρ=0.535, p=4×10^{-16}), Llama-3.1-8B (ρ=0.400, p=5×10^{-9}) and Qwen2.5-Math-7B (ρ=0.340, p=9×10^{-7}), and weakly positive but significant on the prover (ρ=0.192, p=0.007)—while it is null on Qwen3-8B (0.105, n.s.) and Mistral (-0.085, n.s.). That Llama-3.1-8B, a family distinct from Qwen, reproduces the strong per-problem correlation shows the finer per-problem association is not Qwen-specific, though it is still not universal. Qwen2.5-Math-7B's null *asymmetry* arises because both self-generated registers interfere equally (math specialization raised NL interference rather than lowering formal). Hardware note: Qwen2.5-7B was scored on an RTX 4090, the others on an RTX 3090 (bf16); a smoke run reproduced identical NLL values across machines, and all comparisons are within-model.

Prospective held-out test: the ordering does not transfer. The τ=0.87 above is in-sample on the set that produced the metric, so we pre-registered a held-out test before measuring anything (`docs/REGISTER_DISTANCE_PRESPEC_20260809.md`; the original 13-pair binomial was withdrawn in a pre-data amendment after simulation put its true false-positive rate at 0.22–0.34 against a nominal 0.046, and its K=2 design had minimum attainable p=0.50). The replacement primary is an exact permutation test on held-out models *only* — no design pair enters the statistic, because the design pairs are concordant by construction and would carry the result. Protocol, in this order: score r_m on a wide eligible candidate pool; *freeze* five models write-once, chosen on r_m and eligibility alone; check a pre-set separation gate; then measure interference on all five, with no dropping, substitution, or addition.

Eligibility barred every design family (Qwen, DeepSeek, Llama, Mistral), fixed the size band at 6.5–9.5B, and required five *distinct* lineages, since a shared lineage would break the exchangeability the 5!=120 permutation null assumes. Six candidates were scored; the frozen five span r_m 0.113–0.715, a range of 0.602 against the 0.034 gate. As a control on the instrument, the design model Mistral-7B-Instruct was re-scored in the current environment and reproduced its frozen r_m to 1.6×10^{-4}.

Table S14 gives the five frozen models, their r_m, the interference we then measured, and what the design fit predicted.

Table S14: Prospective held-out test of the register-distance ordering. The five models were frozen on r_m before any asymmetry was measured. Under H1 the asymmetry column should increase down the table; it does not. Predictions and intervals come from the frozen design OLS.

Model	Lineage	r_m	asym	predicted	in 95% PI
Yi-1.5-9B-Chat	01-AI	0.113	−0.026	−0.016	yes
Granite-3.3-8B-Instruct	IBM	0.307	−0.015	+0.020	yes
Falcon3-7B-Instruct	TII	0.397	+0.015	+0.037	yes
GLM-4-9B-Chat	Zhipu	0.537	−0.024	+0.063	yes
OLMo-2-7B-Instruct	AI2	0.715	−0.031	+0.096	**no**

The test fails. Four of ten held-out pairs are concordant against a null median of five; the exact one-sided permutation p is 0.76; and the two extreme models are ordered *backwards* — the highest-r_m model, OLMo-2, has the *most negative* asymmetry of the five. Under the pre-specified rule (FAIL at or below the null median) this is a FAIL, not a MIXED result, and we report it as one. Two further observations, both descriptive. First, the sign pattern inverts: 5 of 6 design models had positive asymmetry (formal CoT interfering more), against 1 of 5 held out. Second, half the design set was one family — three of six models were Qwen — while the held-out five are five distinct lineages, so the design association may have been substantially a within-Qwen regularity. The secondary checks, both pre-declared, agree: held-out MAE is 0.056 for the r_m regression against 0.035 for a design-mean baseline (the baseline was pre-recorded as expected to win, from an in-design LOO MAE of 0.0384 vs 0.0270), and the one model outside the frozen 95% prediction interval is the one the regression extrapolates furthest.

What survives is the in-sample measurement itself: register-directed interference is real and large on the models where we measured it, and it does order with r_m within the design set. What does not survive is the reading of that ordering as a predictor. We therefore describe register distance throughout as a measured regularity on the design set, and the pooled 11-model Kendall (τ=0.16) is reported here only as descriptive context — it is not a valid primary, for the same circularity reason the amendment gives. Artifacts: `regdist_heldout_frozen.json` (write-once), `..._asym.csv`, `..._result.json`.

I Boundary Experiment: Lean-Specialized Prover and Second Benchmark

We probe external validity with two open models generated and verified locally (exactly reproducible): a generic 7B (Qwen2.5-7B-Instruct) and the Lean-specialized DeepSeek-Prover-V2-7B, on LeanWorkbook (n=200) and miniF2F-test (n=244). Table S15 reports the generic model; each cell is LLM-contributed proofs / hybrid success rate, with Wilson (Wilson, 1927) 95% intervals on the rate.

This run was re-scored after the audit, and most of it did not survive. The scoring audit of §3 re-scored the panel, Haiku's E6 grid, the E11 LeanWorkbook cells, E9, the transplant cells and the API-error reruns. It did *not* cover the era probes, which remain pre-audit and are therefore not used to establish drift direction or magnitude (§4.1); pass@k was likewise not re-scored, but was re-collected after the fix and its pre-fix numbers are quoted nowhere. It also did *not* cover this boundary run, which was collected in July under the same defective verifier. We therefore re-verified every stored success here under the repaired verifier, re-sending the identical stored tactic text against the identical fixed statement and imports. Of 151 stored LLM successes, 129 were `sorry`- or `admit`-closed and only 22 are sound; 42 of the 129 consisted of the bare token `sorry`, and no cell failed to reproduce for any other reason. Both tables below report the corrected counts, and the contaminated values we previously printed are withdrawn (`scripts/boundary_resorry.py`, `scripts/output/boundary_resorry/`). One limit of this correction: only the `sorry` channel is repaired. The extractor defect of §3 is not, because that would require re-extracting

and re-verifying every boundary cell rather than the stored successes; in the panel that repair *recovered* proofs and recovered them disproportionately outside *Direct* — but the census recovered 36 *Direct* cells too, so the unaudited channel runs both ways. These are therefore `sorry`-corrected results under the original extractor, and the grid **does not establish** the jointly corrected ordering or a bound on its magnitude.

Table S15: Boundary grid, Qwen2.5-7B-Instruct (generic 7B), **re-verified under the repaired verifier** (§3). LLM-contributed proofs / hybrid % (Wilson 95%). On LeanWorkbook both CoT registers stay below *Direct*; on miniF2F every condition falls to 0–1 authored proof once the 29 `sorry`-closed *Direct* successes are removed, so that benchmark supports no ordering for this model. Corrected counts from `boundary_resorry.py`.

Benchmark	Condition	LLM / hybrid %	Wilson 95%
LeanWorkbook	Direct	2 / 22.5%	[17.3, 28.8]
	NL CoT	0 / 21.5%	[16.4, 27.7]
	Lean pseudo	0 / 21.5%	[16.4, 27.7]
miniF2F	Direct	0 / 27.9%	[22.6, 33.8]
	NL CoT	1 / 28.3%	[23.0, 34.2]
	Lean pseudo	0 / 27.9%	[22.6, 33.8]

Truncation caveat. These are reported as *directional*: the CoT conditions truncate more often at the generation token budget than Direct (e.g., Qwen NL CoT on miniF2F: 40/244 truncated vs. 6/244 for Direct), so part of any gap reflects verbose CoT exhausting the budget. After re-verification the generic model retains a *Direct* advantage only on LeanWorkbook (2 against 0 and 0); its miniF2F cells are 0, 1 and 0, which orders nothing, so we withdraw the earlier reading that both CoT registers collapse the contribution on *both* benchmarks. The Lean-specialized DeepSeek-Prover-V2-7B is corroborated at two levels: at the teacher-forced mechanism level (main paper, §Mechanism Measurement) it has the largest register distance (r_m=0.45) and the largest register-directed interference of any model measured; and its generation-protocol boundary grid—re-run with vLLM at a large token budget appropriate to its long subgoal-decomposition outputs, so the CoT conditions no longer truncate—is reported in Table S16. The controlled claim there is only that generic NL and pseudocode scaffolding does not improve on Direct for this specialist, since those conditions share one token budget. The native template's apparent advantage is *withdrawn*, not rescoped: all 70 of its miniF2F and all 6 of its LeanWorkbook stored successes were `sorry`-closed, so after correction it contributes 0 authored proofs on either benchmark and is the weakest condition rather than the strongest. Its outputs are subgoal skeletons of the form `have h : ... := by sorry`, which the defective verifier accepted; we report this as a property of the scoring defect and of that output format, and draw no conclusion about the template's merit, which this run can no longer address.

Table S16: Boundary grid, DeepSeek-Prover-V2-7B (Lean-specialized), generated with vLLM at 4096-token (shared conditions) / 8192-token (native) budgets to avoid the truncation that confounds smaller budgets (NL CoT truncation: 0% vs. 97% at the 1024-token budget). LLM-contributed proofs / hybrid %, **re-verified under the repaired verifier** (§3); corrected counts from `boundary_resorry.py`. The *native* template loses its entire authored contribution on both benchmarks: all 70 miniF2F and all 6 LeanWorkbook stored successes were `sorry`-closed.

Benchmark	Condition	LLM / hybrid %	Wilson 95%
LeanWorkbook	Direct	9 / 26.0%	[20.4, 32.5]
	NL CoT	1 / 22.0%	[16.8, 28.2]
	Lean pseudo	1 / 22.0%	[16.8, 28.2]
	Native	0 / 21.5%	[16.4, 27.7]
miniF2F	Direct	7 / 30.7%	[25.3, 36.8]
	NL CoT	0 / 27.9%	[22.6, 33.8]
	Lean pseudo	1 / 28.3%	[23.0, 34.2]
	Native	0 / 27.9%	[22.6, 33.8]

Prover boundary: the equal-budget ordering survives re-verification, on thinner margins. On the corrected high-budget run both generic-chat CoT registers still sit below the prover's own authored contribution: on miniF2F the LLM-authored count is 7 (Direct) against 0 (NL CoT) and 1 (Lean pseudo), and on LeanWorkbook 9 against 1 and 1. Those are the equal-budget conditions, which share one token budget, and they are the only comparison this run supports. The margins are roughly a third of what we previously printed (21 against 0 and 1; 11 against 1 and 2), because 14 of the stored miniF2F *Direct* successes and 2 of the LeanWorkbook ones were `sorry`-closed. Two readings we previously offered are withdrawn. The native template does *not* reach higher success: all 70 of its miniF2F stored successes carried the `sorry` warning, so its corrected count is 0. And *Direct*'s truncation on this verbose prover ($94/244$ on miniF2F) can no longer be read as making the gap conservative, since two thirds of the *Direct* count it was protecting has been removed; the CoT conditions remain essentially untruncated ($\leq 12/244$). What remains is narrow and one-sided: at equal budget, generic NL and pseudocode scaffolding does not improve on Direct for this specialist. These counts describe the `sorry`-corrected run under the original extractor; they do not establish the ordering, or a bound on the gap, under the jointly repaired pipeline, and they make no claim about the model's native interface.

J Robustness to a Stronger Automation Baseline

Every headline analysis is restricted to the *LLM-contribution regime*: the 402 problems the 16-tactic simple baseline does not solve (500 minus the 98 it does).[2] A reviewer may object that this baseline is understated, inflating the apparent LLM contribution. Two facts bear on this. First, the 16-tactic set already contains the strong single tactics one would name—`aesop`, `omega`, `polyrith`, `nlinarith`, `decide`, `norm_num`, `positivity`, `linarith`, `field_simp`, `simp_all`, `ring`, `native_decide`—so the genuine residual gap is not missing primitives but that the baseline is *single-tactic only*. Second, we test the objection directly: we re-ran an extended baseline of 31 operators (the 16 singles + three further singles + 12 short multi-step combinators such as `simp_all <;> nlinarith` and `constructor <;> nlinarith`) over the same n=500 sample, each operator a single bounded Lean compilation (no hammer such as `duper`/`lean-auto` is installed in this Mathlib, so combinators are the strongest available addition).

The regime is robust. Of the 402 regime problems the strengthened baseline solves only 7 (absorption $7/402 = 1.7\%$, all in one direction since the 31-operator set is a strict superset). The gain is substantively negligible: the strengthened baseline has the *power* to find new solves and finds almost none. A per-solve audit sharpens this — six of the seven are genuine compositional solves and one is a newly added single (`tauto`) — so the absorption is genuine rather than an artifact of nondeterminism, the two nondeterministic cases having already been removed by the problem-level regime definition above. Refitting the primary GLMM on the resulting 395-problem regime leaves every register direction and significance unchanged, with each odds ratio moving by at most 0.012 (Table S17).

[2] The regime is defined at problem level: a problem the baseline solves is excluded for every model. Applying the same rule row-by-row instead yields 404, because on two problems (`lean_workbook_5932`, `lean_workbook_plus_65865`) the baseline succeeds for ten of the twelve models collected and times out for the other two, so a row-level filter lets them re-enter. Both are solved by `nlinarith` and `native_decide` respectively — tactics already in the 16 — so their variation is verifier-timeout nondeterminism, not added power. Earlier drafts of this appendix quoted the row-level 404; every number below uses the problem-level 402 the main text states.

Table S17: Register odds ratios vs. Direct refit on the original 16-tactic regime (402 problems, the problem-level definition the main text uses) vs. the strengthened 31-operator regime (395). Generated by `strong_regime_analysis.py --era era-match --strict`; the left column reproduces the primary GLMM of Table S3 on the re-scored panel.

Register	OR (16-tactic)	OR (31-op)	shift
NL CoT	1.691	1.691	-0.000
Lean pseudocode	1.502	1.507	+0.005
Symbolic math	1.629	1.641	+0.012
Tactic plan	1.179	1.178	-0.001

K MULTI-SAMPLE PASS@k: DO THE EFFECTS SURVIVE SAMPLING?

The main panel is pass@1. To check that the register effects are not a single-sample artifact—we re-ran generation at $N{=}5$ independent samples ($T{=}0.7$) over the LLM-contribution regime ($n{=}157$ problems, seed 42) for two API models whose samples can be re-scored (GPT-4o-mini, Qwen-Plus), verified every sample under the corrected pipeline (§3), and computed pass@k with the unbiased estimator of Chen et al. (2021). Table S18 reports pass@1 and pass@5 per model; the paired variant-minus-Direct deltas with 5,000-replicate bootstrap CIs are in `passk_report.txt`.

Table S18: pass@k over the contribution regime ($N{=}5$, $T{=}0.7$; % of regime problems), re-collected after the scoring fix. "p@1"/"p@5" are pass@1 and pass@5. A third model was sampled but its stored samples cannot be re-scored, so it is excluded. Generated by `passk_analysis.py`.

	GPT-4o-mini		Qwen-plus	
Variant	p@1	p@5	p@1	p@5
Direct	1.3	2.5	7.8	10.2
NL CoT	1.1	2.5	6.6	13.4
Lean psd	1.0	3.2	5.6	10.2
Math sym	2.3	3.2	9.2	14.6
Tactic plan	0.1	0.6	4.2	7.6

The effects do not amplify with k. Re-collected after the scoring fix, this check does not support amplification with decoding budget. At $k{=}5$ *no* register's paired gain over Direct excludes zero on either model (NL CoT $+0.0$pp $[-1.9, +1.9]$ on GPT-4o-mini and $+3.2$pp $[-1.3, +8.3]$ on Qwen-Plus; symbolic math $+0.6$pp and $+4.5$pp, both spanning zero). The only intervals that exclude zero anywhere are tactic plans at $k{=}1$ (-1.1pp $[-2.7, -0.1]$ and -3.6pp $[-7.4, -0.1]$), and tactic plans stay the most negative register at every k. The pass@1-to-pass@5 gap widens in four of eight model×register cells and narrows in the other four. The specific contrast we previously highlighted is gone: Lean pseudocode is now $+0.6$pp $[-1.9, +3.2]$ on GPT-4o-mini and $+0.0$pp $[-5.1, +5.1]$ on Qwen-Plus, so it neither helps one nor hurts the other. On these two models, no paired register gain over Direct is established at $k{=}5$. These numbers are *not* an estimate of the framework's A_m. Pass@k is the probability that at least one of k samples compiles — $1-(1-p)^k$ for a per-sample success probability p under i.i.d. draws — whereas $A_m = \mathbb{E}_z[p_\theta(y^*|z,x)]$ is a single-sample expectation; the former is a monotone function of decoding budget and approaches 1 for any model with positive success probability. We use pass@k only to test whether the *register ordering* persists when additional decoding budget lets the model explore more trajectories. The likelihood object itself is measured by teacher forcing in the main text, not here.

Joint robustness. The stronger-baseline × sampling-budget recomputation was run before the scoring audit, on the uncorrected samples, and we do not carry its numbers forward.

L SECOND BENCHMARK: PER-MODEL CELLS, POWER, AND DIFFERENCE-IN-DIFFERENCES

Full per-cell rates and paired McNemar tests (McNemar, 1947), Holm-corrected across cells (Holm, 1979), on miniF2F-test, the per-model difference-in-differences with bootstrap intervals, and the floor/ceiling and discordant-pair power diagnostics are produced by `e11_analysis.py`: see `e11_report_v3_singlewindow.txt`, `e11_cells_v3_singlewindow.csv`, `e11_did_v3_singlewindow.csv` and `e11_interaction_glmm_v3_singlewindow.csv` in the code appendix.

What the corrected run contains. Both benchmarks were re-collected in one window on the same problem draws, so the benchmark contrast carries no drift term: 1,070 regime rows on miniF2F and 1,755 on LeanWorkbook, 2,825 in the pooled interaction fit. Of the three models sampled, only two can be re-scored (§3), and the two arms are *not* balanced over models: the miniF2F arm carries those two, while the LeanWorkbook arm also carries the third (DeepSeek-V4-Flash, 585 of the 2,825 rows). That model contributes to the benchmark main effect without supplying any within-model benchmark contrast, which is a second reason — beyond the power gate below — to read this interaction as unresolved rather than as an estimate (`e11_pooled_provenance_v4.json`). Direct solves $0/107$ and $4/107$ regime problems, so both endpoints sit at the floor; the median cell has 4 discordant pairs (min 1, max 10) and 7 of 8 fall below ten, which is the power gate's failure condition.

One arm of this contrast had never been re-scored. Both arms are single-window, but only miniF2F was a post-fix re-collection: the LeanWorkbook arm still carried verdicts produced by the defective verifier of §3, so the difference-in-differences was subtracting a repaired arm from an unrepaired one. The re-scored verdicts for those exact cells existed on disk and fed no analysis. We now apply them: coverage is $1{,}755/1{,}755$ rows and 93 verdicts change. Across all three models that arm's solve rate rises from 9.5% to 10.7%, but the rise is one model: DeepSeek-V4-Flash gains 57 extraction recoveries and loses nothing, while the two models that supply every cell of the difference-in-differences table below lose 36 `sorry`-closed successes and gain none, falling from 5.5% to 2.4%. The repair therefore runs in the panel's direction only where the extractor channel dominates; in the cells reported below it is the `sorry` channel that acts, and it removes successes. The uncorrected fit is retained as `*_v4_control` artifacts, which reproduce the previously frozen numbers exactly, so the effect of the repair is auditable rather than asserted.

What the repair changes, and what it does not. It changes the risk-difference contrast: three of the eight difference-in-differences intervals now exclude zero (symbolic math on both models, $+3.7$pp $[0.9, 7.4]$ and $+7.2$pp $[0.7, 14.4]$; tactic plans on GPT-4o-mini, $+3.6$pp $[0.9, 7.9]$) where previously none did, and six cells change sign rather than three. It does not change the pooled inference. Mean-field variational Bayes reports symbolic math $\times$ miniF2F at OR 6.13 $[2.54, 14.79]$, but this appendix has already shown that backend understates posterior variance on this fit (Table S19), so we take the MCMC companion as the estimate: OR 3.48 $[1.19, 10.97]$ corrected, against 2.84 $[1.07, 7.97]$ uncorrected (4 chains, $\hat{R} \leq 1.003$, min ESS 1,699, zero divergences). Exactly one of four interactions excludes 1 either way, and it is the same one. Against a *Direct* rate of 0–3.7% these remain ratios of very small numbers under a failed power gate, so we do not interpret them, and nothing here licenses a statement about whether the register ordering transfers.

The earlier mixed-window decomposition, as provenance. Before the audit we reported a four-arm decomposition on three models, pairing *within* benchmark — each benchmark's two windows share its problem draw — while treating the two benchmarks as distinct problem sets that are never paired with each other. Writing the register effect in an arm as $\Delta_X(v) = \mathrm{rate}_X(v) - \mathrm{rate}_X(\text{Direct})$, the contrasts $A = \Delta_{\text{LWB,Aug}} - \Delta_{\text{LWB,Apr}}$, $B = \Delta_{\text{mF2F,Aug}} - \Delta_{\text{LWB,Aug}}$, $C = \Delta_{\text{mF2F,Jul}} - \Delta_{\text{LWB,Apr}}$ and $D = \Delta_{\text{mF2F,Aug}} - \Delta_{\text{mF2F,Jul}}$ satisfy $C = B + A - D$ exactly on the risk-difference scale, so the drift contamination of the published contrast is $A - D$. That decomposition was computed under the uncorrected scoring pipeline, and on a model set that includes the endpoint we can no longer re-score, so we keep it only to document how this question was previously answered and quote none of its numbers. We also did not report $(A - D)/C$ as a "drift share": C is close to zero for every register, which makes the ratio unstable and its sign uninterpretable.

Baseline integrity across windows. The August arms were verified with a Lean toolchain rebuilt after a machine migration; the April and July arms were not. Both report v4.24.0, which is a version string rather than evidence, so we re-ran the deterministic 16-tactic baseline in the August window and compared problem-ID *sets*, not counts. On miniF2F: $150/150$ problem IDs identical, 43 versus 43 solved, sets equal, zero discordant problems, and the closing tactic identical on all 43 — including the one `nlinarith` and one `native_decide` problem, the only two whose tactics are not deterministic. On LeanWorkbook the April and August baseline-solved sets are likewise identical (33 of 150) for all three models.

Provenance: the withdrawn mixed-window interaction. Our first comparison fit a benchmark×register interaction as an odds ratio across the two original arms, which sat in different collection windows. It is not the reported result, for two independent reasons, and we record it here rather than deleting it so that the change is auditable. First, it estimates C, whose drift component $A - D$ is of the same order as C itself, and whose formal-register values change sign once the third model enters. Second, its intervals were estimator-dependent. Table S19 gives three estimators of that interaction on the same $7{,}293$ rows. The point estimates are stable to within 0.1 on the odds-ratio scale; the intervals are not. Mean-field variational Bayes (`BinomialBayesMixedGLM.fit_vb`, the default in our earlier pipeline) yields intervals about half as wide as MCMC and would report two of four interactions as excluding 1. This is variational underestimation of posterior variance rather than a prior effect: `fit_vb`'s fixed-effect prior is $\mathcal{N}(0, 2)$, more diffuse than the $\mathcal{N}(0, 1)$ used by the MCMC fit, so a prior artifact would widen the MCMC interval, not narrow the variational one. A cluster-robust marginal logit (clustering on problem, models as fixed effects) is a different estimand — marginal rather than conditional odds ratios, hence attenuated toward 1 — and agrees with MCMC on inference. Where an odds ratio was needed we used MCMC, because §4 already declares it the primary analysis, not because it is the most or least favorable; but the reported benchmark result is now the contemporaneous risk-difference contrast B above, which needs no window assumption at all.

Table S19: **The withdrawn mixed-window interaction: point estimates are estimator-robust, intervals are not.** Same model, same rows (n=7,293). "excl. 1" counts interactions whose interval excludes 1. Reported for provenance; superseded by the contemporaneous contrast in Appendix L.

Estimator	Interval	NL CoT	Lean pseudo.	excl. 1
Variational Bayes (mean-field)	95% cred.	0.60 $[0.40, 0.90]$	1.75 $[1.20, 2.57]$	2/4
MCMC (NUTS) — *reported*	95% HDI	0.62 $[0.32, 1.12]$	1.71 $[0.91, 3.09]$	0/4
Cluster-robust marginal logit	95% CI	0.68 $[0.44, 1.06]$	1.56 $[0.97, 2.50]$	0/4

Why the arms are paired. The LeanWorkbook arm of the second collection re-draws its problems with the same loader, seed, and sample-size logic as the main panel, so its problem set is exactly the panel's first 150 problems; the miniF2F arms share the same n=150 seed-42 stratified draw. April and August are therefore paired within problem *and* within register on an unchanged model identifier, which is what makes A a measurement rather than a bound. This bounds the window confound; it does not add power, whose ceiling is set by the smaller miniF2F arm and by three models.

M WITHIN-FAMILY PAIR: FULL GRID

The paired Qwen3-30B ablation summarized in §7.

Table S20: Qwen3-30B paired ablation, re-scored, on the LLM-contribution regime: n=402 five-arm problems for Instruct and n=367 for Thinking after the collection-window policy, with each arm's paired test run on the 378–402 problems the two variants share. Top: Instruct (non-reasoning). Bottom: Thinking (reasoning). Paired McNemar instruct vs. thinking, arm by arm: all $p>0.08$, the closest being tactic plans (20 vs. 10 solves, $p=0.09$).

Variant	Dir.	NL	Psd.	Sym.	Tac.	Δ_{NL}
Instruct (n=402)	4.2%	6.0%	4.2%	3.5%	5.2%	+1.7
Thinking (n=367)	4.4%	7.1%	4.9%	3.3%	3.0%	+2.7

N One-Round Compiler-Feedback Pilot

The main 11-model panel uses one generation and no feedback. To test whether the cheapest workflow repair might change the condition ordering, we ran a *new*, same-window pilot on Haiku 4.5 and MiniMax-M1, selected because the corrected panel gave opposite formal-register directions. Before the new calls, we fixed 24 baseline-unsolved LeanWorkbook problems by a SHA256 seed rule, all five prompt conditions, model settings, one repair for every failed first proof, and paired analysis (`COMPILER_FEEDBACK_PRESPEC_20260920.md`). The model received its first response and up to 1,500 characters of Lean's error; a returned tactic was checked against the same fixed statement. The 240 planned cells all have outcomes: 426 API calls, zero API errors. Provider usage and stop reasons were recorded for each call, unlike in the old panel.

Table S21: One-round compiler-feedback pilot on the same 24 problems per model. First and final columns count verified proofs out of 24. The last column is the paired final success difference from Direct in percentage points (10,000 problem bootstrap draws, 95% interval). This is a new same-window exploratory collection, not a re-analysis of the 11-model panel.

Model	Condition	First	After feedback	Final vs Direct (pp)
Haiku 4.5	Direct	8/24	11/24	reference
Haiku 4.5	NL CoT	9/24	13/24	+8.33 [-8.33, +25.00]
Haiku 4.5	Lean pseudo	12/24	15/24	+16.67 [+0.00, +37.50]
Haiku 4.5	Math symbolic	9/24	13/24	+8.33 [-12.50, +29.17]
Haiku 4.5	Tactic plan	7/24	12/24	+4.17 [-12.50, +20.83]
MiniMax M1	Direct	1/24	2/24	reference
MiniMax M1	NL CoT	3/24	3/24	+4.17 [-8.33, +16.67]
MiniMax M1	Lean pseudo	1/24	3/24	+4.17 [-8.33, +16.67]
MiniMax M1	Math symbolic	1/24	1/24	-4.17 [-12.50, +0.00]
MiniMax M1	Tactic plan	3/24	3/24	+4.17 [-8.33, +16.67]

One round rescued 19 failed Haiku cells and three failed MiniMax cells. Yet every final paired register-minus-Direct bootstrap interval contains zero (Table S21); no arm's improvement in its *contrast* over Direct is resolved on these 24 problems. MiniMax succeeds on only 1–3 problems per arm after feedback, making its ordering particularly uninformative. The two models used 504,364 input and 347,940 output tokens over 426 calls; MiniMax reported 43 length-limit stops among 231 calls (especially on repair calls), while Haiku reported none among 195. This is direct evidence of a token-budget constraint in the new pilot, not an estimate of the old panel's missing stop reasons.

The frozen script has a documented protocol deviation: the pre-call plan said that the repair prompt would print the original imports, but it printed the theorem, previous response and Lean error without explicit imports. Both Lean checks used the original imports, and the same repair prompt format was kept for all arms (`PROTOCOL_DEVIATION.md`). After five cells, a four-worker scheduler continued independent cells without changing the sample, prompts, API parameters or verifier; the scheduling amendment and pre-call script hashes are archived. This small pilot neither establishes that feedback preserves the 11-model average nor that it reverses it. A larger, fully contextualized repair study with matched observed token budgets would be needed for a deployment claim.

O TWO SUPPLEMENTARY ANALYSES ON EXISTING DATA

Both analyses below were pre-registered before either was run (`SUPPLEMENTARY_ANALYSES_PRESPEC_20260921.md`, sha256 `3364a4da...`), which fixed the universe, the normalisation, the strata, the metrics and the falsification condition in advance. Neither makes a provider call: both run on the 21,328 analyzed cells already on disk, and both reproduce the panel's marginal rates exactly (*Direct* 348/4,150, NL CoT 419/3,957), which confirms they use the paper's denominator.

Contamination: no memorization signal in the strata where one would show. LeanWorkbook is public and ships ground-truth proofs, so compiler success could in principle measure recall rather than proof competence. For every cell we re-extracted the model's proof with the repaired extractor and compared it, after a fixed normalisation, with the problem's ground-truth proof (Table S22).

Table S22: Verbatim and near-verbatim (token Jaccard ≥ 0.9) agreement with the LeanWorkbook ground-truth proof, by ground-truth length. The pre-spec excludes the 1-line stratum from any memorization claim in advance: those proofs are canonical single tactics that a competent model may emit independently.

GT length	cells	with response	verbatim	rate	near
1 line	643	636	9	1.42%	9
2 lines	2,808	2,686	2	0.07%	4
3–4 lines	7,805	7,540	**0**	0.00%	17
≥ 5 lines	10,072	9,637	**0**	0.00%	14

In the memorization-indicative strata the verbatim-match count is 0 of 17,177 responses, and near-matches are 31 (0.18%). Matches occur only where the ground-truth proof is one or two lines, and they are not register-skewed (*Direct* 3, tactic plans 4, Lean pseudocode 2, NL CoT 2, symbolic math 0). Five problems drew a verbatim match from any model; success rates on those five are higher (22–31% against 8–12%), which is consistent with their being easy — a one- or two-line canonical proof suffices — rather than with recall, and NL CoT is if anything *lower* there (10.0% against 10.6%). Two limits make every rate a **floor** rather than an estimate: responses are truncated at 4,000 characters, and we compare the extracted proof block rather than the whole response; both can hide a match and neither can invent one. Within those limits we find no evidence that the panel's successes are recalled proofs.

Cost: the one positive result emits 6.42× the characters per success. The runner stored no provider token counts, but it did record each response's untruncated character length, which measures emitted character volume exactly — it is the cost *proxy* that is exact, not the cost. Dividing each register's total emitted characters by its compiler-verified successes gives Table S23:

Table S23: Generation cost per verified success. Characters are exact; approximate tokens use 2.865 characters per token, calibrated from the feedback pilot of Appendix N, which recorded both response text and provider `output_tokens` for the same 426 calls. Generation only: prompt tokens are not recorded per cell, so no full API cost ratio is claimed.

Register	successes	chars/attempt	chars/success	× Direct	≈ tokens/success
Direct	348	147	1,754	1.00	612
NL CoT	419	1,192	11,256	6.42	3,929
Lean pseudocode	509	1,267	10,961	6.25	3,826
Symbolic math	530	1,437	11,942	6.81	4,168
Tactic plans	450	867	8,498	4.84	2,966

NL CoT's panel-average benefit therefore costs 6.42 times Direct's generated characters per verified success. This does not overturn the effect, which is measured per attempt and is unaffected; it qualifies what the effect *recommends*. A practitioner choosing NL CoT buys roughly two extra percentage points of raw success at more than six times the emitted characters, and whether that

trade is worth making is a deployment question this panel cannot answer. Three limits are stated rather than absorbed: this is generation cost only, since prompt tokens are not recorded per cell and the five conditions have different prompt lengths; characters are not tokens, and the calibration comes from two endpoints in one window; and no dollar figure is computed, because provider pricing at collection time was not recorded. Four further analyses on existing data — a negative control for the NL result, generation cost on the second benchmark, the feedback pilot's budget asymmetry, and a contamination instrument for miniF2F — are reported in Appendix P.

P FOUR FURTHER ANALYSES ON EXISTING DATA

These four were pre-registered together, before any of them was run (`SUPPLEMENTARY_ANALYSES_PRESPEC_B_20260922.md`, sha256 `ab24b5a7...`), which fixed each universe, metric, interpretation rule and falsification condition in advance. None makes a provider call. The pre-spec also records, as a disclosure, the two feasibility counts that were already known when it was written, so they cannot later be presented as predictions.

A negative control for the NL result: content, not format or budget. The standing objection to the NL-CoT effect is that it could be a token-budget or formatting artifact rather than a content one. An experiment on disk separates them but had never been scored under the repaired verifier. Its three arms share prompt template, format and budget; only the natural-language content differs. Arm B receives the reasoning generated *for that problem*, arm C receives reasoning generated for a *different*, randomly paired problem.

Table S24: NL negative control, re-verified under the repaired verifier (n=200 LeanWorkbook problems, seed 42, Haiku 4.5). Corrected counts are given as a range because the runner stored each tactic truncated at 200 characters: for a cell at that cap the stored text is not what Lean originally received, so a failure there is uninformative. The range spans the two extreme assumptions. Every unreproducible cell is exactly a capped cell.

Arm	stored	sound	`sorry`	capped	corrected
A. Direct + repair	93	82	7	4	82–86
B. Correct NL + repair	90	82	2	6	82–88
C. Mismatched NL + repair	76	68	5	3	68–71

Of 259 stored successes, 130 are model-produced and 14 of those carry `sorry` or `admit` in the stored tactic text. After re-verification the control *survives*: correct NL beats mismatched NL by +7.0pp, with 16 discordant pairs favouring B against 2 favouring C (exact McNemar p=0.0013), and the correction does not move that contrast (Table S24). This is evidence that the NL *content* carries the effect rather than the extra tokens. Its scope is narrow and we do not widen it: one model, one benchmark, and a repair loop the panel does not have, so the Direct-versus-NL contrast *here* is not a panel replication and is not offered as one.

Cost on the second benchmark, where the measurement is exact. The panel truncated stored *text* at 4,000 characters, so its character counts rest on the separately recorded untruncated length rather than on the stored string. The single-window miniF2F and pass@k collections did not truncate: across all 6,850 stored rows the recorded length equals the stored string length exactly. On miniF2F, *Direct* spends 547 characters per verified success against NL CoT's 8,685 — a factor of 15.9, against the panel's 6.4. Under pass@5 the same ratio is 6.2. The ordering agrees with the panel in every case; the magnitude is benchmark-specific, which is why the main text quotes the panel figure as a panel figure.

The pass@k grid also invites a budget-matched reading — *Direct* at five samples against NL CoT at one — and the pre-spec fixed a 10% tolerance in advance for when that reading is permitted. It is not: the achieved ratio is 0.48, so *Direct* at five samples costs about half of NL CoT at one. We therefore report the raw observation (133,008 characters and 8 solved against 279,276 and 6) and explicitly decline to call it a matched-budget comparison.

What the compiler-feedback repair round costs. The pilot's claim is that a repair round raises success; a repair round is a second API call, so the comparison is budget-asymmetric by construction. Across 240 cells the protocol issues 426 calls (1.78× a one-call protocol) and moves 54 successes to 76, at 443,481 repair tokens — about 20,200 tokens per additional success. This is accounting, and it does not make the pilot a matched-budget experiment.

Contamination on the second benchmark is not instrumentable. The panel's probe compared model output against LeanWorkbook's shipped ground-truth proofs. miniF2F ships none: all 244 statements in the pinned toolchain end in `:= by sorry`. The available substitute is cross-provider agreement — two independent providers emitting a byte-identical normalised proof — and it is weak by construction, since convergence on a canonical tactic looks identical to shared memorization. It finds 68 agreements among 743 comparable miniF2F cells and 38 among 535 under pass@5, but the agreeing proofs are short: 94 of them are one or two tokens, and the longest agreement anywhere is 11 tokens (`linarith` with nine hypotheses). That is the size of a canonical single tactic, not of a recalled proof. With two providers this is under-powered by design, so we report it as *not testable at this power* rather than as evidence of absence.

Q Limitations in Full

What we claim, at which strength. The register-graded hierarchy of §4 is the observational core; reasoning-label moderation remains unresolved, the reference-risk ROPE probabilities describing that uncertainty rather than establishing general non-moderation; the presentation effect is *withdrawn* — after re-scoring neither the equal-rewrite design nor a pre-registered independent renderer establishes it — while the cost of *translating* reasoning between registers survives the equal-rewrite control but is not reproduced by the independent renderer's cells, and the pilot's consistency interaction is retracted as translation damage (§6); register distance is a retrospective association on the six design models and *not* a predictor — a pre-registered held-out test on five new lineages rejected the ordering (§5); and generation-confidence shifts (Appendix G) and per-model case studies remain exploratory. Appendix H gives the per-model correlations and family breakdown behind the register-distance proxy.

The main text summarizes the limitations below; the complete statements follow. The prospective held-out check of the register-distance ordering has now been run and failed (Appendix H); what would still sharpen the account is localizing which token classes carry the interference, and establishing whether any cheap model-only measurement predicts a model's CoT response, since this one does not. Lean 4's syntax (released 2023) is plausibly under-represented in pretraining relative to NL mathematics, which is the pretraining-composition reading of the register-distance gradient; we state it as an interpretation, since we cannot inspect any of these models' training mixtures.

(1) The 11-model main panel is pass@1, so its small per-cell differences are paired, pooled directions, not calibrated effect sizes. On the two models whose panel samples were re-scorable (GPT-4o-mini, Qwen-Plus) we re-collected generations after the scoring fix at $N{=}5$, $T{=}0.7$ and computed pass@k (Chen et al., 2021) with paired bootstrap CIs. At $k{=}5$ *no* register's paired gain over Direct excludes zero on either model; the only intervals that do are tactic plans at $k{=}1$ (-1.1pp $[-2.7, -0.1]$ and -3.6pp $[-7.4, -0.1]$). The gap between pass@1 and pass@5 widens in four of eight model×register cells and narrows in the other four, so we withdraw the claim that these effects amplify with decoding budget; two models cannot settle it either way (Appendix K). Note that pass@k is the probability that at least one of k samples compiles, not the single-sample quantity A_m of §2; we use it to test whether the register ordering persists when additional decoding budget marginalizes over more sampled trajectories, not as an estimator of A_m. (2) The presentation effect is *withdrawn*, not rescoped: after re-scoring neither the equal-rewrite design (OR 1.14 $[0.91, 1.43]$) nor the pre-registered independent renderer (OR 1.02 $[0.86, 1.21]$) establishes it, whose subject models are distinct from its renderer. One model, Qwen-Plus, does show a large presentation advantage under that renderer (2.89 $[1.59, 5.24]$), which we report as further heterogeneity rather than as a surviving effect. Independently of the outcome, the design cannot separate "presentation register" from residual stylistic properties of the renderings. (3) Effects could partly reflect prompt wording rather than register per se. Our paraphrase-robustness check uses three same-register wordings each for NL and Lean-pseudocode CoT; re-scored, it shows no sign reversal. On the one model with a complete re-scored grid (Haiku 4.5, $n{=}157$ paired problems, $n{=}47$ for the third Lean wording), no NL

wording reverses direction (nets $+5$, $+6$, $+0$ against Direct) and the two quotable Lean-pseudocode wordings point *positive* ($+10$, p=0.03; $+6$). The third is *withheld rather than reported*: 70.1% of its attempts returned API errors, leaving n=47, and the audit's own $> 5\%$ rule excludes that arm from quotation — so although its re-scored net is also positive, we do not count it. The earlier claims in both directions — that an NL paraphrase flips the sign, and that the Lean-pseudocode penalty is stable across wordings — were artifacts of the scoring defects of §3. One model and two registers cannot establish paraphrase robustness in general, so we claim only that we no longer have evidence of a wording reversal. (4) Compiler success is a thresholded proxy for the likelihood object A_m in our framework, and the per-problem register-distance association is significant in four of six models and replicates across two families (Qwen, Llama) but is not universal, and the aggregate cross-model ordering that does hold across all six design models is in-sample only: a pre-registered test on five held-out lineages did not reproduce it ($4/10$ concordant pairs, p=0.76; Appendix H). (5) The generation-based boundary run of the Lean-specialized DeepSeek-Prover-V2-7B (Ren et al., 2025) was collected under the defective verifier and fell outside the audit's re-scoring scope; we re-verified it, and 129 of its 151 stored successes were `sorry`-closed (Appendix I). After correction the equal-budget ordering survives on both benchmarks (9 against 1 and 1; 7 against 0 and 1) but at roughly a third of the printed margin, the prover's native-template advantage is withdrawn entirely ($70 \to 0$ authored proofs), and the generic 7B's miniF2F cells fall to 0–1 and order nothing. Because only the `sorry` channel is repaired and the extractor channel is not in either condition, that grid establishes no jointly corrected ordering. The prover is separately corroborated by the teacher-forced mechanism measurement (§5), which uses likelihoods rather than compiler verdicts and is therefore untouched by this defect, and where it exhibits the largest register distance and interference. (6) Dynamic API endpoints limit exact reproducibility, and we report the probe that motivated our window policy: paired re-runs of previously clean cells four months later moved DeepSeek-V4-Flash's NL-CoT accuracy $+12.8$pp with its Direct accuracy statistically unchanged (§4.1). Those probe cells were scored before the audit of §3 and have not been re-scored, so they establish neither the direction nor the magnitude of post-correction drift. Because the drift is register-specific it does not cancel in a register contrast, which is why every model is held to a single collection window. One further endpoint, Kimi-K2-Thinking, became unavailable mid-study with 27 cells unrepairable and is excluded whole-row. The supplementary material records model snapshots, collection dates per cell, decoding parameters, all prompts, and per-problem outcomes.

R REPRODUCIBILITY

Models and endpoints. The 11 analyzed API endpoints, plus the two excluded rows (Kimi-K2-Thinking and DeepSeek-R1), are listed in `model_snapshots.csv` in the code appendix with provider, requested model identifier, and the completion date of each source file. API-returned snapshot strings were not recorded, so the requested identifier is the finest version information available. Pre-audit temporal probes motivated the window policy of §4.1: because they have not been re-scored they establish neither the direction nor the magnitude of post-correction drift, and every model is held to the single collection window defined there.

Decoding. API models: temperature 0.7; `max_tokens` is set per provider by the runner — 4,096 for OpenAI, Anthropic, MiniMax and Zhipu endpoints and 16,384 for the DeepSeek and Qwen endpoints, whose reasoning variants return thinking tokens. The stored result files do not record the value, so we report the runner's configuration rather than a per-cell attestation. Open-weight teacher-forced runs: greedy CoT generation, deterministic scoring, seed 42.

Data. LeanWorkbook sample: n=500 (observational) and n=200 (transplant/teacher-forced), both seed 42; the problem set is MIT-licensed and ships with the supplement. The simple-tactic baseline attempts 16 single tactics before any LLM call. Responses were stored truncated at a 4,000-character cap, which is why 1,205 of 21,328 cells (5.65%) cannot be re-scored. Truncation is strongly register-dependent — 0.39% of *Direct* rows against 4.15% of tactic plans, 6.65% of NL CoT, 7.36% of Lean pseudocode and 9.51% of symbolic math — and it is also concentrated by model (DeepSeek-Chat 34%; Haiku and Qwen3-thinking 0%). §3 bounds what that missingness can do to the estimates. The supplement ships all prompts (5 variants, 4 rewrite templates, judge rubric), per-problem outcomes carrying both the original and the corrected verdict, and the stored response

text for every cell the audit re-verified. It does *not* ship the full raw result tree, and generation cannot be re-run from it, because the API endpoints have since drifted (§4.1).

Verification. Lean 4 v4.24.0 with Mathlib pinned at `f897ebcf72cd16f89ab4577d0c826cd14afaafc7` (tag `v4.24.0`), via `lake env lean`, with a 60 s wall-clock timeout per attempt; compiler errors are read from stdout (Lean writes diagnostics there, not to stderr). Every problem inherits `set_option maxHeartbeats 0` from the LeanWorkbook preamble, so the elaborator is *unbounded* and the 60 s clock is the only limit — the verifier's own `maxHeartbeats` default never applies on this path. The toolchain recorded here is the one that performed the re-scoring; per-run toolchain versions were not captured at collection time, so we attest the re-verification rather than the original collection. An executable local control compiles a fixed true statement, rejects a restated weaker theorem inserted inside a false target's tactic block, and rejects `sorry`; `#print axioms` reports no axioms for the positive control (`statement_integrity_probe.py`). This tests the harness boundary, not every generated proof. The original API runner stored neither `finish_reason` nor token usage, so provider-side length stops and per-arm token costs cannot be reconstructed; the per-arm storage-cap counts above are a different quantity. A fixed-seed local diagnostic selected three recovered successes per register (15 total): all recompiled, and `#print axioms` showed no `sorryAx` (`recovered_axioms_audit.json`). This sample is a check on axiom leakage, not a bound for all recovered proofs.