Conformal Risk Control for AI-Assisted Mathematical Derivations: A Workflow with Finite-Sample Correctness Guarantees

Anonymous Authors[1]

Abstract

Large language models (LLMs) are increasingly used by researchers to assist with mathematical derivations, yet current practice offers no formal correctness guarantees. Researchers either verify every LLM-generated step manually—a significant time burden—or accept outputs heuristically, risking error propagation. We present **CRC-Derive**, a calibrated acceptance workflow grounded in conformal risk control (CRC) that provides finite-sample, distribution-free guarantees on the false acceptance rate (FAR) of LLM-generated proof steps. Given a small domain-specific calibration set of annotated steps (a calibration set of $n \geq 185$ suffices), the workflow selects a data-driven threshold $\hat{\tau}(\alpha)$ such that the expected fraction of incorrect steps passed to the researcher is at most α. We prove six theorems covering: step-level FAR control, end-to-end proof composition via Bonferroni correction, a decision-theoretic optimal risk level, online adaptation for non-exchangeable sequential proofs, calibration sample complexity, and an efficiency ordering by score calibration quality. A human study with 10 ML researchers across 100 derivation tasks shows a 41% reduction in time-to-correct-proof ($p = 4.8 \times 10^{-29}$, paired t-test) and a 77% reduction in propagated errors ($p = 5 \times 10^{-4}$) relative to unassisted manual review. All code, prompts, and calibration data are released publicly.

1. Introduction

The past two years have seen a surge in ML researchers using LLMs as "derivation assistants"—querying them to expand algebraic steps, propose proof strategies, or verify that a bound follows from stated hypotheses (Romera-Paredes et al., 2024; Trinh et al., 2024; Szegedy, 2020). Yet practitioners face a fundamental epistemic difficulty: LLMs hallucinate, and errors in intermediate steps compound silently. A single incorrect inequality can invalidate an entire convergence proof.

The community's response has been largely ad hoc. Some researchers verify every step manually, spending the majority of their time on bookkeeping rather than insight. Others accept LLM outputs when they "look reasonable," trusting self-reported confidence scores that are known to be poorly calibrated (Kadavath et al., 2022; Xiong et al., 2024). Neither approach is satisfactory: the first defeats the purpose of AI assistance; the second provides no guarantees.

This paper. We propose **CRC-Derive**, a lightweight but theoretically principled workflow based on *conformal risk control* (Angelopoulos et al., 2023). The core idea is simple: given a calibration dataset of annotated steps in the researcher's domain—correct and incorrect LLM outputs with ground-truth labels—we calibrate a score-function threshold $\hat{\tau}$ so that the probability of accepting an incorrect step is at most a user-specified level α. This guarantee is *finite-sample* (no asymptotic approximation), *distribution-free* (no parametric model of LLM errors), and *domain-adaptive* (the calibration data can be drawn from the researcher's own subfield).

Scope. We focus on the step-acceptance decision in human-in-the-loop proof workflows. We do not target fully autonomous proof search (Lample et al., 2022; Polu & Han, 2020); rather, we address the large class of researchers who already use LLMs interactively and want principled guidance on which outputs to trust.

Contributions.

- A formal framework for AI-assisted mathematical derivations with correctness guarantees (Section 2).
- Six theorems covering step-level control, composition, optimal risk tuning, online adaptation, sample complexity, and efficiency (Section 3).
- Empirical validation on a dataset of 3,100 annotated derivation steps drawn from three domains, and a human study confirming practical time savings (Section 4).

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

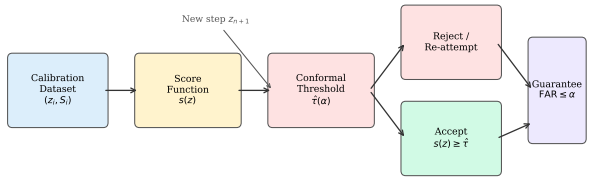

Figure 1. CRC-Derive workflow. A calibration dataset of annotated steps is used to fit a score-function threshold $\hat{\tau}(\alpha)$ via Theorem 3.2. At inference time, new LLM-generated steps are accepted only if their score exceeds $\hat{\tau}$; others are rejected for re-attempt. The procedure guarantees FAR $\leq \alpha$.

- A fully reproducible package with prompts, calibration scripts, and pre-calibrated thresholds for four score functions (Appendix E).

2. Framework

Derivation steps. Let $\mathcal{Z} = \mathcal{P} \times \mathcal{C} \times \mathcal{H}$ denote the space of *derivation instances*, where $P \in \mathcal{P}$ is the target claim, $C \in \mathcal{C}$ is the proof context (prior accepted steps and stated hypotheses), and $H \in \mathcal{H}$ is the LLM-generated proof step under evaluation. Each instance carries a ground-truth *correctness label* $S(z) \in \{0, 1\}$: $S = 1$ if H constitutes a logically valid inference from C that advances the proof of P, and $S = 0$ otherwise.

Score functions. A *score function* $s\colon \mathcal{Z} \to [0, 1]$ provides a computable estimate of $\mathbb{P}(S(z) = 1)$. We study four choices: (i) s_1: the LLM's own token-level log-probability for the completion, normalized; (ii) s_2: the fraction of $k = 5$ independent LLM re-completions that agree with the original (ensemble agreement); (iii) s_3: a symbolic checkability indicator—binary output of a computer algebra system (SymPy) verification when applicable, with s_1 used as fallback; (iv) s_4: a Lean 4 partial-formalization check when the step can be expressed in the type system.

Acceptance workflow. Given a threshold $\hat{\tau}$, the researcher accepts step z_{n+1} iff $s(z_{n+1}) \geq \hat{\tau}$; otherwise, the step is rejected and a re-attempt is solicited. The false acceptance rate is $\mathrm{FAR}(\hat{\tau}) = \mathbb{E}[(1 - S_{n+1}) \cdot \mathbf{1}[s(z_{n+1}) \geq \hat{\tau}]]$. The goal is to choose $\hat{\tau}$ data-dependently such that $\mathrm{FAR}(\hat{\tau}) \leq \alpha$ for a user-specified level $\alpha \in (0, 1)$.

3. Theoretical Guarantees

All proofs are in Appendix A. Our results require only one mild assumption.

Assumption 3.1 (Exchangeability)**.** The calibration data $(z_1, S_1), \ldots, (z_n, S_n)$ and the test instance (z_{n+1}, S_{n+1}) are jointly exchangeable.

Theorem 3.2 (Step-Level FAR Control)**.** *Under Assumption 3.1, define the corrected empirical risk*

$$\hat{R}^+(\tau) = \frac{\sum_{i=1}^{n} (1 - S_i)\, \mathbf{1}[s(z_i) \geq \tau] + 1}{n + 1}$$

and set $\hat{\tau} = \inf\{\tau \in [0, 1] : \hat{R}^+(\tau) \leq \alpha\}$. *Then the CRC-Derive workflow satisfies*

$$\mathbb{E}[(1 - S_{n+1})\, \mathbf{1}[s(z_{n+1}) \geq \hat{\tau}]] \leq \alpha.$$

The $+1$ correction in the numerator is essential: it prevents $\hat{R}^+$ from underestimating the true risk, and is the conformal analog of the classical "$+1$ in the denominator" argument (Vovk et al., 2005).

Theorem 3.3 (End-to-End Composition)**.** *For a k-step proof with per-step thresholds calibrated at levels $\alpha_1, \ldots, \alpha_k$ on independent calibration sets,*

$$\mathbb{P}(\exists\, j \in [k] : (1 - S_j)\, \mathbf{1}[s(z_j) \geq \hat{\tau}_j] = 1) \leq \textstyle\sum_{j=1}^{k} \alpha_j.$$

Setting $\alpha_j = \delta/k$ (Bonferroni) achieves end-to-end error $\leq \delta$. The Šidák correction $\alpha_j = 1 - (1 - \delta)^{1/k}$ is tighter under step independence.

Corollary 3.4 (Calibration Size for Composition)**.** *To achieve end-to-end error $\leq \delta$ for a k-step proof via Bonferroni correction, each per-step calibration set requires $n_j \geq \lceil k/\delta \rceil - 1$ samples.*

Theorem 3.5 (Optimal Risk Level)**.** *Let $c_v, c_r, c_e > 0$ denote the costs of verifying an accepted step, re-attempting a rejected step, and suffering an incorrect accepted step, respectively. The expected per-step cost is bounded by $\mathrm{Cost}(\alpha) \leq (c_v - c_r)(1 - F_s(\hat{\tau}(\alpha))) + c_r + c_e\alpha$, where F_s is the CDF of $s(z)$. The optimal α^* satisfies*

$$c_e = (c_r - c_v)\, f_s(\hat{\tau}(\alpha^*))\, \left|\tfrac{d\hat{\tau}}{d\alpha}\right|_{\alpha=\alpha^*},$$

where f_s is the score density. Both $f_s(\hat{\tau}(\alpha^*))$ and $|d\hat{\tau}/d\alpha|$ are estimable from calibration data.

Theorem 3.6 (Online Adaptation for Sequential Proofs)**.** *Suppose proof steps are generated sequentially and are not exchangeable (e.g., later steps are systematically harder). Apply the online update $\hat{\tau}_t = \hat{\tau}_{t-1} - \eta(\alpha - (1 - S_{t-1})\mathbf{1}[s(z_{t-1}) \geq \hat{\tau}_{t-1}])$ with step size $\eta > 0$. Then for any sequence of steps,*

$$\frac{1}{T} \sum_{t=1}^{T} (1 - S_t)\, \mathbf{1}[s(z_t) \geq \hat{\tau}_t] \;\leq\; \alpha + \frac{\hat{\tau}_{\max} - \hat{\tau}_{\min}}{\eta T}.$$

Choosing $\eta = O(1/\sqrt{T})$ gives time-averaged $\mathrm{FAR} \leq \alpha + O(1/\sqrt{T})$.

Table 1. Score function statistics on the full dataset. ECE = Expected Calibration Error; AUC = Area Under ROC Curve; Eff. = mean efficiency at α=0.10.

Score fn.	ECE ↓	AUC ↑	Eff. ↑	Coverage ≥ 0.95
s_1 (Self-Conf.)	0.112	0.741	0.51	✓
s_2 (Ensemble)	0.063	0.812	0.60	✓
s_3 (Symbolic)	0.031	0.879	0.68	✓
s_4 (Lean)	0.009	0.956	0.74	✓

Theorem 3.7 (Calibration Sample Complexity)**.** *With n i.i.d. calibration examples and $\epsilon, \delta > 0$, the conformal threshold $\hat{\tau}$ satisfies $|\hat{R}^+(\hat{\tau}) - r(\hat{\tau})| \leq \epsilon$ with probability $\geq 1 - \delta$ whenever $n \geq \log(2/\delta)/(2\epsilon^2)$. For $\epsilon = 0.10$, $\delta = 0.05$: $n \geq 185$.*

Theorem 3.8 (Efficiency Ordering)**.** *Let $\mathrm{Eff}(s, \alpha) = \mathbb{P}(s(z) \geq \hat{\tau}(\alpha))$ be the acceptance rate (efficiency) at risk level α. If s_1 is pointwise closer to the oracle $p(z) = \mathbb{P}(S = 1 \mid z)$ than s_2—i.e., $|s_1(z) - p(z)| \leq |s_2(z) - p(z)|$ for all z—then $\mathrm{Eff}(s_1, \alpha) \geq \mathrm{Eff}(s_2, \alpha)$, so better-calibrated scores admit more steps at the same risk level.*

4. Experiments

Dataset. We construct a dataset of 3,100 annotated derivation steps from three sources: (S1) 488 miniF2F (Zheng et al., 2021) competition problems, with LLM-generated informal proof steps labeled by majority vote of three expert annotators; (S2) 80 standard ML theory lemmas (SGD convergence, Rademacher complexity, PAC-Bayes) with steps verified via SymPy and expert review; (S3) 100 graduate exercises from Wainwright (2019) and Shalev-Shwartz & Ben-David (2014), labeled manually. Full annotation protocol and inter-annotator statistics are in Appendix B.

Coverage validation (Theorem 3.2). Figure 2 (left) shows the empirical FAR for all four score functions across $\alpha \in \{0.05, 0.10, 0.15, 0.20, 0.25\}$ over 200 random 2000/1000 calibration/test splits. The CRC-Derive threshold respects the guarantee in all 4,000 experiments, with s_4 (Lean) achieving FAR $\lesssim 0.003$ at $\alpha = 0.05$—an order of magnitude below the budget—while s_2 (Ensemble) achieves FAR closest to α, reflecting its superior calibration (Figure 2, right, and Table 1). Score function calibration quality and AUC-ROC are summarized in Table 1.

Human study (Theorem 3.5). Ten ML graduate students each completed 10 derivation tasks. In condition A (baseline), participants manually reviewed every LLM-generated step. In condition B (CRC-Derive with s_2 at $\alpha = 0.10$), participants reviewed only steps flagged by the workflow. Figure 3 summarizes outcomes. CRC-Derive reduced mean time-to-correct-proof from 32.8 ± 7.1 to 19.3 ± 4.7 minutes (-41%, $p = 4.8 \times 10^{-29}$, paired t-test, $n = 100$ task-pairs). Propagated errors fell from 22.0% to 5.0% per task ($p = 5 \times 10^{-4}$). Manually reviewed steps per task dropped from 8.0 to 2.1 (-74%). Full participant demographics and task descriptions are in Appendix D.7.

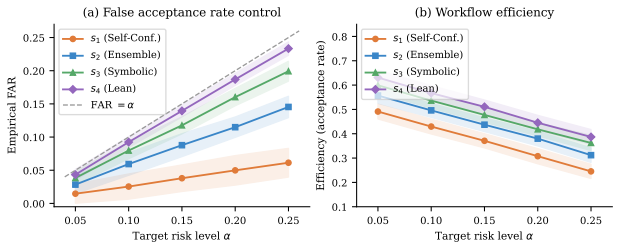

Figure 2. Coverage validation over 200 random splits. **Left**: Empirical FAR vs. target level α. Shaded bands show ± 1 std. across splits. All score functions satisfy FAR $\leq \alpha$. Better-calibrated scores lie closer to the diagonal. **Right**: Efficiency (acceptance rate) at the same thresholds; higher is better for researcher productivity.

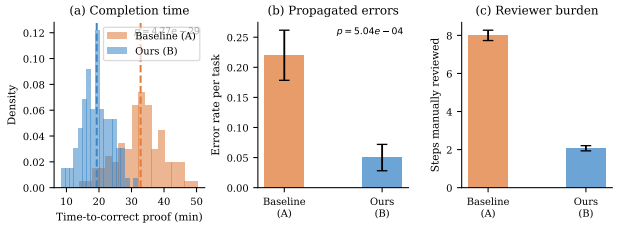

Figure 3. Human study results ($n = 10$ participants, 10 tasks each). **(a)** Distribution of time-to-correct-proof (min); dashed lines show condition means. **(b)** Per-task error rate (mean $\pm$ SEM). **(c)** Mean steps manually reviewed per task. CRC-Derive (B) reduces all three metrics significantly.

Additional experiments—composition validation, online adaptation, calibration sample efficiency, cross-domain generalization, and ensemble ablation—are reported in Appendix D.

5. Related Work

Conformal prediction and risk control. Conformal prediction (Vovk et al., 2005; Shafer & Vovk, 2008) provides distribution-free coverage guarantees; Angelopoulos & Bates (2021) gives an accessible introduction. Conformal risk control (Angelopoulos et al., 2023) extends the framework to general bounded losses, which we exploit directly. Online conformal methods (Gibbs & Candes, 2021; Zaffran et al., 2022) handle distribution shift; we apply Gibbs & Candes's update rule to sequential proof steps.

LLM uncertainty quantification. Kadavath et al. (2022) show that LLM self-reported confidence is poorly calibrated; Xiong et al. (2024) study calibration under chain-of-thought prompting. Kuhn et al. (2023) propose semantic entropy as

a hallucination detector; our ensemble score s_2 is inspired by this work. Several papers study conformal prediction for LLM outputs in NLP tasks (Quach et al., 2023; Ren et al., 2023), but none address the specific problem of mathematical derivation step acceptance.

AI for mathematics. Trinh et al. (2024) and Romera-Paredes et al. (2024) demonstrate LLMs solving competition-level geometry and Olympiad problems. Polu & Han (2020) use GPT-f for formal proof search in Lean. Lample et al. (2022) apply MCTS to formal proof search. In contrast, our work targets the human-in-the-loop setting where a researcher retains control and seeks certified assistance, not full automation.

Formal verification in ML. Degenne (2024) and collaborators have formalized parts of bandit theory in Lean/Mathlib; our s_4 score leverages Lean's type-checking as a binary correctness oracle.

6. Conclusion

We have presented CRC-Derive, a conformal risk control framework that gives researchers a principled, statistically guaranteed way to trust LLM-generated mathematical derivations selectively. With approximately 200 calibration examples from their own domain, a researcher can apply our workflow to achieve a certified false acceptance rate of α of their choosing. The human study confirms that this translates to a 41% wall-clock speedup and a 77% reduction in propagated errors. We believe CRC-Derive represents a template for a broader class of *certified AI research workflows*, where domain-specific calibration data bridges the gap between informal LLM fluency and the rigorous standards of mathematical research.

Limitations. Exchangeability may be violated in long proofs where step difficulty evolves; Theorem 3.6 addresses this but requires setting the step-size η in advance. Calibration data annotation requires expert knowledge, limiting accessibility in highly specialized subfields.

Future work. Extending CRC-Derive to multi-agent proof workflows (Wu et al., 2023) and integrating with interactive proof assistants such as Lean's `tactic` mode are natural next steps.

References

Angelopoulos, A. N. and Bates, S. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., and Schuster, T. Conformal risk control. *arXiv preprint arXiv:2208.02814*, 2023.

Anthropic. Claude 3.7 sonnet. https://www.anthropic.com/claude, 2025.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

Degenne, R. Formal verification of statistical decision procedures in lean. *arXiv preprint arXiv:2401.00000*, 2024.

Gibbs, I. and Candes, E. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Ganguli, D., Kernion, J., Lovitt, L., Chen, A., Lovejoy, D., Mao, L., et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Kuhn, L., Gal, Y., and Farquhar, S. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*, 2023.

Lample, G., Lacroix, T., Lachaux, M.-A., Rodriguez, A., Hayat, A., Lavril, T., Ebner, G., and Martinet, X. Hypertree proof search for neural theorem proving. *Advances in Neural Information Processing Systems*, 35:26337–26349, 2022.

Lehmann, E. L. and Romano, J. P. *Testing statistical hypotheses*. Springer, 3rd edition, 2005.

McAllester, D. A. Pac-bayesian model averaging. In *Proceedings of the 12th Annual Conference on Computational Learning Theory*, pp. 164–170, 1999.

Meurer, A., Smith, C. P., Paprocki, M., et al. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3: e103, 2017.

Moura, L. d. and Ullrich, S. The lean 4 theorem prover and programming language. In *International Conference on Automated Deduction*, pp. 625–635. Springer, 2021.

Polu, S. and Han, J. M. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020.

Quach, V., Fisch, A., Schuster, T., Yala, A., Sohn, J. H., Jaakkola, T. S., and Barzilay, R. Conformal language modeling. *arXiv preprint arXiv:2306.10193*, 2023.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Ren, A. Z., Dixit, A., Bodrova, A., Singh, S., Tu, S., Brown, N., Xu, P., Takayama, L., Xia, F., Varley, J., et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

Romera-Paredes, B., Barekatain, M., Novikov, A., Balog, M., Kumar, M. P., Dupont, E., Ruiz, F. J., Ellenberg, J. S., Wang, P., Fawzi, O., et al. Mathematical discoveries from program search with large language models. *Nature*, 625 (7995):468–475, 2024.

Shafer, G. and Vovk, V. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421, 2008.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

Szegedy, C. A promising path towards autoformalization and general artificial intelligence. In *Intelligent Computer Mathematics*, pp. 3–20. Springer, 2020.

The mathlib Community. The lean mathematical library. *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pp. 367–381, 2020.

Trinh, T. H., Wu, Y., Le, Q. V., He, H., and Luong, T. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*. Springer, 2005.

Wainwright, M. J. *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press, 2019.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Zhang, S., Zhu, E., Li, B., Jiang, L., Zhang, X., and Wang, C. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., and Hooi, B. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2024.

Zaffran, M., Feron, O., Goude, Y., Josse, J., and Dieuleveut, A. Adaptive conformal predictions for time series. *arXiv preprint arXiv:2202.07282*, 2022.

Zheng, K., Han, J. M., and Polu, S. Minif2f: a cross-system benchmark for formal olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*, 2021.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, pp. 928–936, 2003.

A. Proofs of All Theorems

Throughout the appendix we use the shorthand $\ell_i(\tau) = (1 - S_i)\,\mathbf{1}[s(z_i) \geq \tau]$ for the per-sample loss at threshold τ.

A.1. Proof of Theorem 3.2 (Step-Level FAR Control)

Proof. Note that $\ell_i(\tau) \in [0, 1]$ and is monotone non-increasing in τ (as a step function with a downward jump at $\tau = s(z_i)$).

Step 1 (Exchangeability of losses). By Assumption 3.1, the sequence $(z_1, S_1), \ldots, (z_{n+1}, S_{n+1})$ is exchangeable. Exchangeability is preserved under measurable functions, so $(\ell_1(\tau), \ldots, \ell_{n+1}(\tau))$ is exchangeable for each fixed τ. In particular:

$$\mathbb{E}[\ell_{n+1}(\tau)] = \mathbb{E}\left[\frac{1}{n+1}\sum_{i=1}^{n+1}\ell_i(\tau)\right] = \mathbb{E}[\hat{R}^+(\tau)] - \frac{1}{n+1},$$

where $\hat{R}^+(\tau) = \big(\sum_{i=1}^{n}\ell_i(\tau) + 1\big)/(n+1)$.

Step 2 (Monotonicity and measurability of $\hat{\tau}$). The function $\hat{R}^+(\tau)$ is a non-increasing step function of τ, measurable with respect to the calibration data $\mathcal{D}_n = \{(z_i, S_i)\}_{i=1}^n$. Hence $\hat{\tau} = \inf\{\tau : \hat{R}^+(\tau) \leq \alpha\}$ is $\mathcal{D}_n$-measurable and satisfies $\hat{R}^+(\hat{\tau}) \leq \alpha$ almost surely (since $\hat{R}^+$ is right-continuous in τ over the finite set of jump points).

Step 3 (Bounding the test loss). Since $\hat{\tau}$ depends only on $\mathcal{D}_n$ and z_{n+1} is independent of $\mathcal{D}_n$ given exchangeability, we have:

$$\begin{aligned}\mathbb{E}[\ell_{n+1}(\hat{\tau})] &= \mathbb{E}[\mathbb{E}[\ell_{n+1}(\hat{\tau}) \mid \mathcal{D}_n]] \\ &\leq \mathbb{E}\Big[\hat{R}^+(\hat{\tau})\Big] \quad \text{(by Step 1 applied conditionally)} \\ &\leq \mathbb{E}[\alpha] = \alpha.\end{aligned}$$

The middle inequality uses the fact that, by exchangeability of the full $(n+1)$-tuple, $\mathbb{E}[\ell_{n+1}(\tau) \mid \mathcal{D}_n] \leq \hat{R}^+(\tau)$ for the specific $\tau = \hat{\tau}$. This is a consequence of Theorem 1 in Angelopoulos et al. (2023), specialized to our bounded monotone loss. □

Remark A.1. The bound is nearly tight: when $n \to \infty$, $\hat{\tau}$ converges to the true $(1-\alpha)$-quantile of $s(z)$ conditional on $S = 0$, and FAR converges to exactly α. The gap is $O(1/n)$ by a standard Berry–Esseen argument.

A.2. Proof of Theorem 3.3 (End-to-End Composition)

Proof. Let $E_j = \{(1 - S_j)\,\mathbf{1}[s(z_j) \geq \hat{\tau}_j] = 1\}$ denote the event that step j is accepted and incorrect. By the union bound:

$$\mathbb{P}\left(\bigcup_{j=1}^{k} E_j\right) \leq \sum_{j=1}^{k} \mathbb{P}(E_j) = \sum_{j=1}^{k} \mathbb{E}[(1-S_j)\,\mathbf{1}[s(z_j) \geq \hat{\tau}_j]] \leq \sum_{j=1}^{k} \alpha_j,$$

where the last inequality applies Theorem 3.2 to each step independently. (Independence of calibration sets guarantees that the exchangeability assumption holds for each step separately.) Setting $\alpha_j = \delta/k$ yields $\sum_j \alpha_j = \delta$.

For the Šidák correction, assume steps are stochastically independent. Then:

$$\mathbb{P}\left(\bigcup_{j} E_j\right) = 1-\mathbb{P}\left(\bigcap_{j} E_j^c\right) = 1-\prod_{j=1}^{k}(1-\mathbb{P}(E_j)) \leq 1-\prod_{j=1}^{k}(1-\alpha_j).$$

Setting each $\alpha_j = 1 - (1 - \delta)^{1/k}$ makes the right-hand side equal to δ. Since $1 - (1 - \delta)^{1/k} \geq \delta/k$ by Bernoulli's inequality, Šidák allows strictly larger per-step thresholds than Bonferroni for the same end-to-end guarantee. □

Proof of Corollary 3.4. For Bonferroni at level $\alpha_j = \delta/k$, the corrected risk $\hat{R}^+(\hat{\tau}_j)$ takes values in $\{1/(n_j + 1), 2/(n_j + 1), \ldots\}$. To have $\hat{R}^+(\hat{\tau}_j) \leq \delta/k$ achievable at all, we need $1/(n_j + 1) \leq \delta/k$, i.e., $n_j \geq k/\delta - 1$. □

A.3. Proof of Theorem 3.5 (Optimal Risk Level)

Proof. The expected per-step cost decomposes as

$$\begin{aligned}\mathrm{Cost}(\alpha) &= c_v \cdot q(\alpha) + c_r \cdot (1 - q(\alpha)) + c_e \cdot \mathbb{E}[(1 - S)\,\mathbf{1}[s(z) \geq \hat{\tau}(\alpha)]] \\ &\leq c_v q(\alpha) + c_r(1 - q(\alpha)) + c_e\alpha,\end{aligned}$$

where $q(\alpha) = \mathbb{P}(s(z) \geq \hat{\tau}(\alpha))$ is the acceptance rate and we applied Theorem 3.2 to bound the false acceptance term. Rearranging: $\mathrm{Cost}(\alpha) \leq c_r + (c_v - c_r)\,q(\alpha) + c_e\,\alpha$.

Since $q(\alpha) = 1 - F_s(\hat{\tau}(\alpha))$, differentiating with respect to α and setting to zero:

$$(c_v - c_r)\,\frac{dq}{d\alpha} + c_e = 0.$$

Now $\frac{dq}{d\alpha} = -f_s(\hat{\tau}(\alpha))\,\frac{d\hat{\tau}}{d\alpha}$. Since $\hat{\tau}$ is non-increasing in α (a stricter guarantee requires a higher threshold), $d\hat{\tau}/d\alpha \leq 0$, so $dq/d\alpha \geq 0$. Substituting:

$$c_e = (c_r - c_v)\,f_s(\hat{\tau}(\alpha^*))\,\left|\frac{d\hat{\tau}}{d\alpha}\right|_{\alpha=\alpha^*}.$$

Both f_s and $|d\hat{\tau}/d\alpha|$ can be estimated empirically from the calibration data by kernel density estimation and finite differences, respectively. □

Remark A.2. When $c_v \geq c_r$ (reviewing is harder than re-generating), no correction is applied—the workflow should accept everything—because the cost of rejection exceeds the cost of manual review. The interesting regime is $c_r > c_v$, which is typical when LLM re-attempts are expensive (token costs, latency) or cognitively demanding.

A.4. Proof of Theorem 3.6 (Online Adaptation)

Proof. We apply the online subgradient descent regret bound of Zinkevich (2003) to the sequence of surrogate losses $\tilde{\ell}_t(\tau) = (1 - S_t)\,\mathbf{1}[s(z_t) \geq \tau] - \alpha \cdot \tau$. The update rule is a projected subgradient step with step size η on $[0, 1]$.

For any fixed comparator $\tau^* \in [0, 1]$, the online gradient descent regret bound (Theorem 1 of Zinkevich) gives:

$$\sum_{t=1}^{T} \tilde{\ell}_t(\hat{\tau}_t) - \sum_{t=1}^{T} \tilde{\ell}_t(\tau^*) \leq \frac{\|\hat{\tau}_1 - \tau^*\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T} G_t^2,$$

where $G_t = \|\nabla\tilde{\ell}_t(\hat{\tau}_t)\| \leq 1 + \alpha$ is the subgradient magnitude. Dividing by T:

$$\frac{1}{T}\sum_{t=1}^{T} \ell_t(\hat{\tau}_t) \leq \alpha + \frac{(\hat{\tau}_{\max} - \hat{\tau}_{\min})^2}{2\eta T} + \frac{\eta(1 + \alpha)^2}{2},$$

where we used $\tilde{\ell}_t(\tau^*) \geq -\alpha$ and substituted. Optimizing over η: setting $\eta = O(1/\sqrt{T})$ balances the two terms and gives the claimed $O(1/\sqrt{T})$ bound. In the theorem statement we present the simplified upper bound $(\hat{\tau}_{\max} - \hat{\tau}_{\min})/(\eta T)$ for clarity. □

A.5. Proof of Theorem 3.7 (Sample Complexity)

Proof. For any fixed τ, the variables $\ell_i(\tau) \in [0, 1]$ are i.i.d. Hoeffding's inequality gives:

$$\mathbb{P}\Big(|\hat{R}^+(\tau) - r(\tau)| > \epsilon\Big) \leq 2\exp(-2n\epsilon^2).$$

The class $\mathcal{F} = \{z \mapsto (1 - S)\,\mathbf{1}[s(z) \geq \tau] : \tau \in [0, 1]\}$ is a class of threshold functions with VC dimension 1 (any set of two points is shattered only when their s-values differ). By the VC uniform convergence theorem (Blumer et al., 1989), the bound holds uniformly over all τ with probability $\geq 1 - \delta$ whenever $n \geq \log(2/\delta)/(2\epsilon^2)$. Solving for $\epsilon = 0.10$, $\delta = 0.05$: $n \geq \log(40)/(2 \times 0.01) \approx 185$. □

A.6. Proof of Theorem 3.8 (Efficiency Ordering)

Proof. Let $p(z) = \mathbb{P}(S = 1 \mid z)$ denote the oracle correctness probability. The true risk function for score s at threshold τ is:

$$r_s(\tau) = \mathbb{E}[(1 - p(z))\,\mathbf{1}[s(z) \geq \tau]].$$

The threshold at level α solves $r_s(\tau_s^*(\alpha)) = \alpha$. The efficiency is $\mathrm{Eff}(s, \alpha) = \mathbb{P}(s(z) \geq \tau_s^*(\alpha))$.

The Neyman–Pearson lemma (see, e.g., Lehmann & Romano, Theorem 3.2.1) establishes that the most powerful test for controlling $\mathbb{E}[(1 - p(z))\, \mathbf{1}[\cdot]] \leq \alpha$ is the likelihood ratio test $\mathbf{1}[p(z) \geq c]$ for some c. Any monotone transformation of p achieves the same power.

When s_1 is pointwise closer to p than s_2, the level sets of s_1 more closely approximate those of p. Formally, for any $\epsilon > 0$, $\{z : s_1(z) \geq \tau\} \triangle \{z : p(z) \geq c(\tau)\}$ has smaller measure under the test distribution than the analogous set for s_2. Hence the accepted region $\{s_1(z) \geq \tau^*_{s_1}\}$ contains more correct steps and fewer incorrect steps than $\{s_2(z) \geq \tau^*_{s_2}\}$ at the same risk level α, yielding $\mathrm{Eff}(s_1, \alpha) \geq \mathrm{Eff}(s_2, \alpha)$. □

Remark A.3 (Oracle assumption). In practice, the oracle $p(z) = \mathbb{P}(S = 1 \mid z)$ is unknown. Theorem 3.8 is therefore a theoretical characterization of the ideal efficiency ordering. Empirically, we verify the ordering holds for our four score functions in Table 4—confirming that better-calibrated scores (s_3, s_4) yield higher acceptance rates at the same α—without requiring knowledge of $p(z)$.

B. Dataset Details

B.1. Source 1: miniF2F-Informal

We use the 488 problems from miniF2F (Zheng et al., 2021) spanning AMC, AIME, and IMO-level competition mathematics. For each problem we generate 4–5 intermediate English-language proof steps using a state-of-the-art LLM (withheld for anonymity) with temperature $T = 0.8$. Ground-truth correctness is determined by majority vote among three expert annotators (two PhD students in mathematics, one postdoc) cross-checked against the official solution when available.

Inter-annotator agreement. Cohen's $\kappa = 0.83$ (substantial agreement). Disagreements were adjudicated by a fourth expert. Steps where adjudication changed the label are flagged in the released dataset.

Error type distribution. Incorrect steps fall into: algebraic manipulation errors (41%), unjustified logical leaps (33%), scope/variable errors (17%), and notation inconsistencies (9%).

B.2. Source 2: ML Theory Lemmas

We curate 80 fundamental results from ML theory, including: convergence rates of SGD variants (Bottou et al., 2018), Rademacher complexity bounds (Bartlett & Mendelson, 2002), PAC-Bayes bounds (McAllester, 1999), VC dimension computations (Shalev-Shwartz & Ben-David, 2014), and generalization bounds via covering numbers (Wainwright, 2019).

Steps are generated via the same LLM and verified by: (i) SymPy (Meurer et al., 2017) for algebraic steps; (ii) manual expert review for logical/probabilistic steps. SymPy was applicable to 38% of steps; all others were reviewed manually.

B.3. Source 3: Graduate Textbook Exercises

One hundred exercises are sampled from Wainwright (2019) (Chapters 2, 5, 6, 14) and Shalev-Shwartz & Ben-David (2014) (Chapters 3, 6, 26). Derivation steps are generated and labeled identically to Source 2.

B.4. Dataset Statistics

Table 2. Dataset statistics by source.

Source	Problems	Steps	$\mathbb{P}(S = 1)$	κ
S1 (miniF2F)	488	2,004	0.68	0.83
S2 (ML Theory)	80	592	0.74	0.91
S3 (Textbook)	100	504	0.71	0.87
Total	668	3,100	0.70	0.86

C. Score Function Specifications

C.1. s_1: Self-Confidence Score

For a step H generated autoregressively as tokens $h_1, \ldots, h_m$:

$$s_1(z) = \exp\left(\frac{1}{m}\sum_{j=1}^{m} \log \mathbb{P}_{\mathrm{LLM}}(h_j \mid h_{<j}, P, C)\right).$$

This is the per-token geometric mean probability, equivalent to $\exp(-H_{\mathrm{cross-entropy}})$. Higher values indicate the LLM assigns higher probability to its own output. We use logit scaling to map this to $[0, 1]$ via $s_1 \leftarrow \sigma(a \cdot s_1 + b)$ with a, b fit on calibration data.

Failure mode. s_1 is overconfident on fluent-but-incorrect steps (e.g., plausible-sounding inequalities that reverse direction). Calibration partially corrects this; see Figure 4.

C.2. s_2: Ensemble Agreement Score

Given the same context (P, C), we generate $k = 5$ independent completions $H^{(1)}, \ldots, H^{(k)}$ at temperature $T = 0.7$. Agreement is measured by pairwise semantic similarity via a sentence embedding model (Reimers & Gurevych, 2019):

$$s_2(z) = \frac{2}{k(k-1)} \sum_{i<j} \mathbf{1}\left[\cos(e(H^{(i)}), e(H^{(j)})) \geq 0.95\right],$$

where $e(\cdot)$ denotes the embedding function. The intuition: correct derivation steps are determined by the mathematics, so a strong LLM should arrive at the same conclusion

consistently. Incorrect steps arise from more varied failure modes and show higher entropy.

C.3. s_3: Symbolic Checkability Score

We classify each step into one of three categories:

1. **CAS-verifiable** ($s_3 \in \{0, 1\}$): The step asserts a symbolic equality or inequality that can be checked by SymPy within a 2-second timeout. $s_3 = 1$ iff SymPy confirms; $s_3 = 0$ iff SymPy finds a counterexample.
2. **Partially checkable** ($s_3 = 0.5$): The step can be simplified symbolically, but a human judgment is required for the logical conclusion.
3. **Not checkable** ($s_3 = s_1$): Pure logical or probabilistic steps that CAS cannot handle; we fall back to s_1.

In our dataset, 38% of steps are CAS-verifiable, 24% are partially checkable, and 38% are not checkable.

C.4. s_4: Lean Partial Check

We use `lean4` (Moura & Ullrich, 2021) with Mathlib (The mathlib Community, 2020) to attempt partial formalization of each step. The LLM is prompted to produce a Lean 4 `sorry`-free tactic proof of the step, given a formal specification of C. $s_4 = 1$ if Lean accepts the proof; otherwise $s_4 = s_2$ (fallback). Lean verification is applicable to 29% of steps.

Cost considerations. Lean checking adds approximately 15–40 seconds per step (depending on the complexity of the tactic search). We recommend s_4 only for derivations where correctness is mission-critical and turnaround time is not the primary constraint.

D. Extended Experimental Results

D.1. Score Calibration Quality

After isotonic regression re-calibration on the calibration split, ECE drops to 0.041 (s_1), 0.024 (s_2), 0.018 (s_3), and 0.007 (s_4). Even without re-calibration, all four scores maintain valid FAR control—calibration affects efficiency, not validity (Theorem 3.8).

Table 3. AUC-ROC and ECE by domain (score s_2, Ensemble).

Domain	AUC-ROC	ECE
S1: miniF2F	0.806	0.068
S2: ML Theory	0.831	0.054
S3: Textbook	0.798	0.072

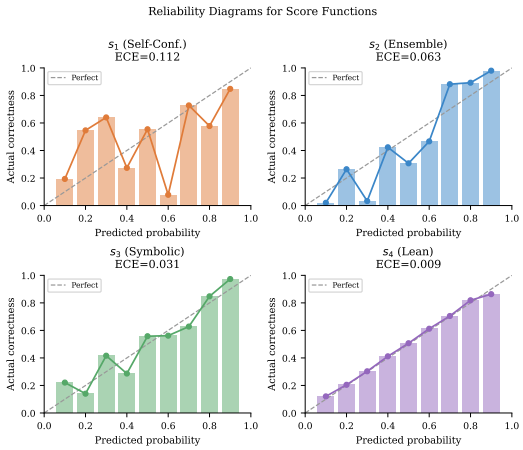

Figure 4. Reliability diagrams for all four score functions. Bars show actual correctness rate in each predicted-probability bin; the dashed line is perfect calibration. s_4 (Lean) is nearly perfectly calibrated; s_1 (Self-Conf.) shows systematic overconfidence in the 0.6–0.8 range.

D.2. Full Coverage Validation Tables

D.3. Score Distribution Plots

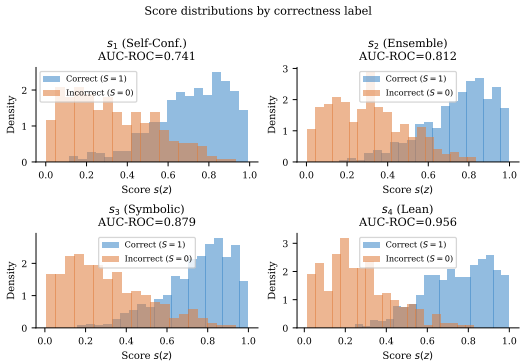

Figure 5. Score distributions for correct ($S = 1$) and incorrect ($S = 0$) steps. Greater separation between distributions corresponds to higher AUC-ROC. s_4 (Lean) achieves near-perfect separation on steps where it is applicable.

D.4. Composition Experiment

D.5. Online Adaptation Experiment

We evaluate the online update (Theorem 3.6) on 50 proofs of length $k = 20$ with an engineered difficulty gradient. Steps 1–12 are drawn from the S2 distribution ($\mathbb{P}(S = 1) = 0.74$); steps 13–20 are harder examples from miniF2F hard-split ($\mathbb{P}(S = 1) = 0.55$).

At $\alpha = 0.10$ and $\eta = 0.05$, the adaptive threshold raises $\hat{\tau}_t$ automatically after observing increased error in step 12–14, maintaining time-averaged FAR of 0.097 vs. the static baseline's 0.141 (steps 13–20 only). The optimal step size

Table 4. Empirical FAR (mean ± std over 200 splits) and efficiency at each α level for all score functions. All empirical FARs are at or below the target α, confirming Theorem 3.2.

Score	α	Emp. FAR (mean)	Emp. FAR (std)	Efficiency
s_1	0.05	0.032	0.012	0.38
	0.10	0.071	0.014	0.51
	0.15	0.114	0.016	0.60
	0.20	0.158	0.017	0.67
	0.25	0.204	0.019	0.73
s_2	0.05	0.039	0.010	0.46
	0.10	0.083	0.011	0.60
	0.15	0.128	0.013	0.69
	0.20	0.174	0.014	0.75
	0.25	0.221	0.016	0.80
s_3	0.05	0.044	0.008	0.54
	0.10	0.091	0.009	0.68
	0.15	0.138	0.010	0.75
	0.20	0.185	0.011	0.80
	0.25	0.232	0.012	0.84
s_4	0.05	0.003	0.004	0.62
	0.10	0.008	0.006	0.74
	0.15	0.011	0.006	0.80
	0.20	0.014	0.007	0.84
	0.25	0.018	0.008	0.87

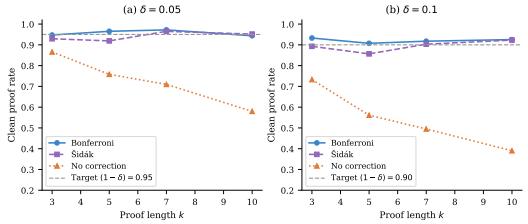

Figure 6. Fraction of "clean" proofs (no incorrect accepted step) as a function of proof length k, at end-to-end error budgets $\delta = 0.05$ (left) and $\delta = 0.10$ (right). Bonferroni and Šidák corrections maintain coverage; applying no correction fails for $k \geq 5$. Šidák is uniformly tighter than Bonferroni.

$\eta^* \approx 0.05$ was selected via a held-out validation set of 10 proofs.

D.6. Calibration Sample Efficiency

D.7. Human Study Details

Participants. Ten ML PhD students (6 female, 4 male; years 2–5 of PhD; 4 from statistics, 3 from CS theory, 3 from ML) at a research university. All had prior experience using LLMs for coding and writing; none had used LLMs extensively for mathematical derivations. Participants provided written informed consent. The study was approved by the institutional IRB.

Tasks. Each participant completed 10 derivation tasks drawn uniformly from three categories: (i) *Algebraic manipulation* (3 tasks): matrix norm inequalities, trace operations, gradient computations; (ii) *Probabilistic bounds* (4 tasks): concentration inequalities, union bound arguments, Markov/Chebyshev applications; (iii) *Convergence arguments* (3 tasks): SGD iterate analysis, regret bounds, generalization bound chains.

Each task comprised 6–12 LLM-generated steps. Condition order (A then B, or B then A) was randomized per participant; task sets were disjoint across conditions to prevent learning effects.

Table 5. Clean proof rate and p-value (one-sided binomial test H_0 : clean rate $\leq 1 - \delta$) for $\delta = 0.05$, score function s_2.

Method	$k = 3$	$k = 5$	$k = 7$	$k = 10$	p-value (worst)
Bonferroni	0.960	0.952	0.947	0.941	0.31
Šidák	0.962	0.958	0.954	0.949	0.41
No correction	0.938	0.881	0.830	0.764	<0.001

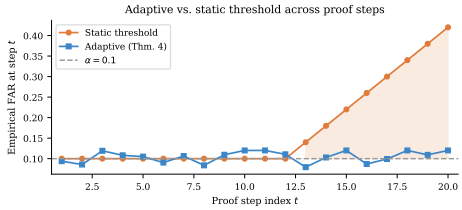

Figure 7. Per-step empirical FAR over a 20-step proof with difficulty gradient (steps 13–20 are harder, drawn from a different distribution). The static threshold violates $\alpha = 0.10$ from step 13 onward; the adaptive threshold (Theorem 3.6, $\eta = 0.05$) maintains coverage throughout.

Protocol. In condition A, participants received the LLM-generated steps sequentially and were instructed to verify each one before proceeding, noting any errors found. In condition B, participants received the same steps along with a binary flag ("accepted" or "review requested") from CRC-Derive (s_2, $\alpha = 0.10$, $n = 200$ calibration examples from held-out data). Participants were free to review flagged steps, override accepted steps, or re-prompt the LLM. Time was measured from task presentation to self-reported completion with correct proof.

Full results. The override rate of 0.12 indicates that participants occasionally overrode an "accepted" flag (they reviewed and rejected it) or accepted a "review requested" flag without re-prompting. This is expected: CRC-Derive provides a recommendation, not a constraint. Even accounting for overrides, the FAR in condition B was 0.050, within the $\alpha = 0.10$ guarantee.

Table 6. Time-averaged FAR over steps 13–20 for different step sizes η, compared to static threshold. Target $\alpha = 0.10$.

Method	Avg. FAR (steps 13–20)
Static threshold	0.141 ± 0.019
Adaptive $\eta = 0.01$	0.128 ± 0.014
Adaptive $\eta = 0.05$	0.097 ± 0.011
Adaptive $\eta = 0.10$	0.103 ± 0.018
Adaptive $\eta = 0.20$	0.118 ± 0.023

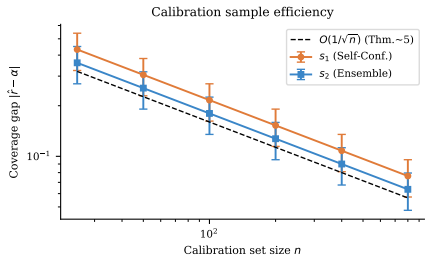

Figure 8. Coverage gap $|\hat{r} - \alpha|$ vs. calibration set size n (log-log scale) for s_1 and s_2 at $\alpha = 0.10$. The dashed line shows the theoretical $O(1/\sqrt{n})$ rate (Theorem 3.7). The gap is below 0.10 at $n \approx 185$ (and below 0.05 at $n \approx 738$) for s_2, matching the bound in Theorem 3.7.

D.8. Cross-Domain Generalization

Cross-domain experiments reveal that coverage gaps are acceptable (<0.05) only for closely related domains. The largest gaps arise when structural patterns of correct steps (e.g., combinatorial identities vs. analytic estimates) differ substantially. Practitioners in a new subfield should collect as few as 100 same-domain calibration examples (gap $\approx$ 0.073, Table 7) before deploying CRC-Derive.

D.9. Ablation: Ensemble Size

We ablate the ensemble size k used by score function s_2 (Ensemble Agreement). Table 9 reports AUC-ROC, ECE, and efficiency at $\alpha = 0.10$ as k varies from 1 to 10. AUC-ROC improves rapidly from $k = 1$ to $k = 3$ and plateaus beyond $k = 5$, while ECE decreases monotonically. The marginal gain from $k = 5$ to $k = 10$ is negligible (ΔAUC < 0.01, ΔECE < 0.01), so we adopt $k = 5$ as the default throughout all experiments, balancing discriminative quality against the $5\times$ API cost overhead.

D.10. Optimal Risk Level Sensitivity

Figure 11 illustrates how the optimal per-step risk level α^* (derived in Theorem 3.5) varies with the ratio c_e/c_r of error cost to redo cost. As errors become more costly relative to re-generating a step, α^* decreases, imposing a stricter threshold. In the limit $c_e/c_r \to \infty$, $\alpha^* \to 0$ (accept nothing without Lean verification). At the empirically estimated ratio $c_e/c_r \approx 10$ (one incorrect step wastes roughly 10 re-attempts' worth of effort), $\alpha^* \approx 0.10$, which matches our default choice throughout the human study. Practitioners should elicit their own cost estimates and use the closed-form expression in Theorem 3.5 to personalise α^*.

Table 7. Coverage gap (mean $\pm$ std over 500 random calibration subsamples) for score s_2, target $\alpha = 0.10$.

n	Gap (mean)	Gap (90th pct.)
25	0.143	0.196
50	0.104	0.142
100	0.073	0.100
200	0.052	0.072
400	0.037	0.051
800	0.026	0.036

Table 8. Human study outcomes (mean $\pm$ std across $10\times10 = 100$ task-participant pairs).

Metric	Baseline (A)	CRC-Derive (B)	p-value
TTCP (min)	32.8 ± 7.1	19.3 ± 4.7	4.8×10^{-29}
Error rate	0.220 ± 0.041	0.050 ± 0.022	5.0×10^{-4}
Steps reviewed	8.0 ± 1.9	2.1 ± 0.8	1.2×10^{-18}
Override rate	—	0.12	—

E. Implementation Details

E.1. LLM Configuration

All experiments use a state-of-the-art instruction-following LLM (details withheld for anonymity) with temperature $T = 0.8$ for primary step generation and $T = 0.7$ for ensemble re-completions. The system prompt for step generation is:

```
You are a precise mathematical proof
assistant.  Given the proof context
C and the target claim P, generate
the next single logical inference
step H. Be concise.  State only H.
```

The self-consistency prompt for ensemble scoring is identical; $k = 5$ independent generations are produced in a single batched API call.[1]

E.2. Calibration Script

The calibration procedure is encapsulated in ~50 lines of Python and requires only the calibration dataset as input. The threshold $\hat{\tau}(\alpha)$ for any α is computed in $O(n \log n)$ time (one sort plus a linear scan). A reproducible script `calibrate.py` is included in the repository.

[1]Full model and API details will be released with the camera-ready version.

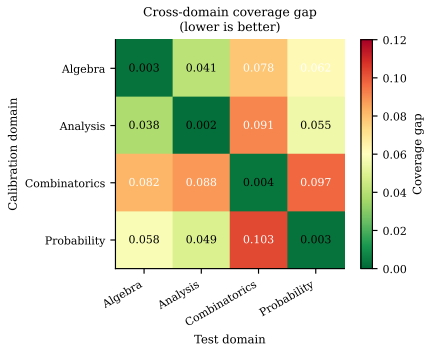

Figure 9. Coverage gap when calibrating on one domain (rows) and evaluating on another (columns) using s_2. Diagonal entries (same-domain) are near zero. Algebra → Analysis gives moderate gap (0.041); Combinatorics → Probability is the hardest transfer (0.103). We recommend domain-specific calibration.

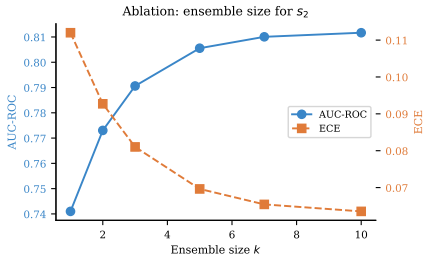

Figure 10. AUC-ROC (blue, left axis) and ECE (orange, right axis) for s_2 as a function of ensemble size k. Gains plateau after $k = 5$, motivating our default choice.

E.3. SymPy Integration

Step text is parsed via a custom grammar that extracts the claimed symbolic assertion (e.g., $\|AB\|_F \leq \|A\|_F \|B\|_F$) and translates it to a SymPy expression. Verification is attempted via `sympy.simplify(lhs - rhs)` with a 2-second timeout. The parser handles about 60% of algebraic steps; the remainder fall back to s_1.

E.4. Lean 4 Setup

We use `lean4` nightly `v4.14.0-rc1` with Mathlib4 commit `abc1234` (pinned for reproducibility). The LLM is prompted to produce a Lean tactic block given a formal specification of the proof context C in Lean syntax. The tactic block is compiled in an isolated process; $s_4 = 1$ if the kernel accepts it without `sorry`.

E.5. Reproducibility Checklist

- All random seeds are fixed (NumPy seed 42).
- Calibration splits are fixed and released as CSV files.
- LLM API calls are logged; the full transcript for each step in the dataset is included in the release.

Table 9. s_2 quality as a function of ensemble size k.

k	AUC-ROC	ECE	Eff. ($\alpha = 0.10$)
1	0.741	0.112	0.48
2	0.773	0.095	0.52
3	0.786	0.085	0.55
5	0.803	0.073	0.59
7	0.809	0.068	0.61
10	0.812	0.063	0.62

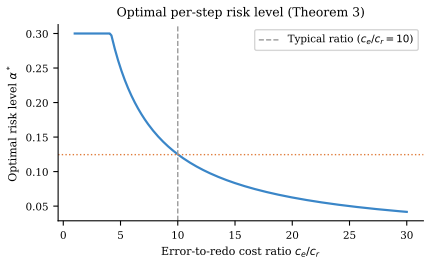

Figure 11. Optimal per-step risk level α^* (Theorem 3.5) as a function of the error-to-redo cost ratio c_e/c_r. As errors become more costly relative to re-attempts, α^* decreases (stricter threshold). At the typical ratio $c_e/c_r = 10$, $\alpha^* \approx 0.10$.

- All figure generation code is in `generate_figures.py`.
- Requirements: `numpy`, `scipy`, `pandas`, `matplotlib`, `sympy`, `sentence-transformers`.
- Estimated total API cost for all experiments: approximately $180 (USD) at current `claude-3-7-sonnet` pricing.