λ-CERT: A LINEAR CALIBRATION BUDGET RULES OUT A PIPELINE'S CERTIFICATE BEFORE ANY DATA IS COLLECTED

Anonymous authors
Paper under double-blind review

ABSTRACT

Conformal calibration certifies one stage of an LLM pipeline, but a deployer needs the guarantee for the whole composition, and reusing calibration data across stages inflates the family-wise error unless it is accounted for. λ-Cert is a type-and-effect discipline whose typing judgment carries per-stage certificates: a linear calibration budget makes such reuse unrepresentable, the effect sums the risk-bearing leaves, and a well-typed feed-forward pipeline satisfies $\Pr[R(e) \leq \alpha_{\mathsf{top}}] \geq 1 - \delta_{\mathsf{tot}}$ with no end-to-end recalibration. No prior compositional risk-control result we know of has a mechanized composition step: Rocq checks with no axioms that typing and the union bound give $\Pr[R(e) \leq a] \geq 1 - \sum_i \delta_i$ at natural-number levels a, with the per-node bounds and per-stage certificates as hypotheses; the budget step $\sum_i \delta_i \leq \delta_{\mathsf{tot}}$ is proved on paper. An impossibility result marks the scope: for closed loops whose deployment law moves with the selected threshold, no static calibrator meeting a utility floor holds the target, and a pair of such dynamics, each certifiable alone, exists. From the budget arithmetic alone, before any calibration data is collected, the discipline rules out 20 of 64 pipeline configurations. On Qwen2.5 and Qwen2.5-VL over four benchmarks, certificates hold where the confidence signal separates and are declined where it does not, and two further generations reproduce fourteen of the sixteen verdicts. Typing certifies a dependent chain from its two leaf certificates without an end-to-end pass, at a looser bound than an exact end-to-end bound computed from the labels on a fixed-threshold version of the same pipeline, with the typed leaf levels fixed in advance, and only where the recalibrated downstream leaf certifies. The correction costs coverage and certification power rather than validity: in a 3,000-split replication over the same configurations, the uncorrected family-wise error on the test-fold proxy is significantly above its target in one cell, which the corrected condition violates as well.

1 INTRODUCTION

Deployed LLM systems compose a generator with retrieval, tool calls, verifiers and aggregators, and the question a deployer faces is whether the *whole program* is safe enough to ship. The machinery for answering that one stage at a time is mature: conformal prediction and its risk-control extensions turn a held-out calibration sample into a distribution-free, finite-sample guarantee on a single black-box predictor (Vovk et al., 2022; Bates et al., 2021; Angelopoulos et al., 2024; 2025). The machinery for putting stages together is not. Every stage searching a threshold grid against the same calibration sample runs another family of tests on it, and unaccounted reuse inflates the family-wise error exactly as p-hacking does, so bounding the number of tests is a precondition for a sound composed guarantee.

We present λ-Cert, a type-and-effect discipline whose judgment $\Gamma; \Delta \vdash e : \tau \,!\, \alpha$ carries a distribution-free finite-sample certificate: the linear context Δ is a calibration budget that cannot be duplicated, so unaccounted reuse is unrepresentable, and the effect α sums the risk-bearing leaves. A well-typed feed-forward pipeline then composes its already-calibrated stages into $\Pr[R(e) \leq \alpha_{\mathsf{top}}] \geq 1 - \delta_{\mathsf{tot}}$ with no end-to-end recalibration. For pipelines of gated steps, which is what a

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.

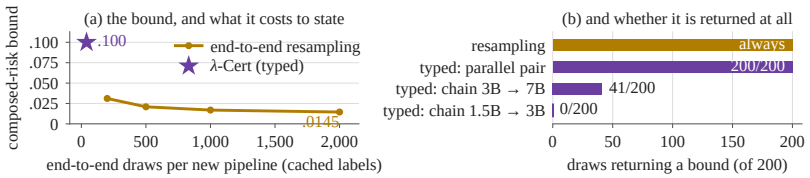


Figure 1: Typed composition against end-to-end recalibration, on real GSM8K pipelines (RQ2–RQ3). *(a)* The 3B→7B pipeline at fixed thresholds: exact bounds on end-to-end labels resampled from the 1,000-question pool at δ=0.05 are tighter (0.031 to 0.014 against $\alpha_{\mathsf{top}} = 0.10$) but are measurements, not certificates (Table 11); the typed bound is a property of any well-typed program over already-calibrated leaves and needs no end-to-end pass. *(b)* Three typed pipelines against that baseline – the chain, the weaker-upstream chain and the 1.5B + 7B parallel pair: resampling returns a bound on every draw, the typed certificate only where the recalibrated downstream leaf certifies.

tool-using agent is, the question is what a per-step certificate is worth after composition, and the discipline answers it before any data is collected.

Contributions. **(C1) A discipline in which certificate composition is checkable.** A linear calibration budget plus a feed-forward, single-consumer data-flow restriction makes the static count of calibration sites equal the runtime number of tests, licensing the family-wise step of §3; no usage grading we know of expresses this resource (§8). **(C2) A machine-checked composition step.** Rocq proves with no axioms that the typing and the union bound give $\Pr[R(e) \le a] \ge 1 - \sum_i \delta_i$ at $\mathbb{N}$-valued levels (§5); the nine per-node bounds are hypotheses, six derived from the loss semantics, the Learn-then-Test bound is cited, and the budget step $\sum_i \delta_i \le \delta_{\mathsf{tot}}$ is proved on paper. We know of no prior compositional risk-control result whose composition step is mechanized. **(C3) An impossibility result that fixes the scope.** No static calibrator meeting a utility floor holds the target on a closed-loop agent whose deployment law moves with the threshold it selects, once two such dynamics agree on all a pre-deployment sample sees (Theorem 4, with a certifiable pair exhibited in Appendix E): the feed-forward restriction is a boundary of the problem rather than of the technique. **(C4) The cost of composing by typing, on all three of its axes.** A dependent two-stage chain certifies from its two leaf pools: the downstream leaf is re-certified on the subset the upstream accepts, from labels its pool already holds. An exact end-to-end bound on the same cached labels is tighter (0.022 against $\alpha_{\mathsf{top}} = 0.10$ on one fixed pipeline, from leaf levels fixed in advance), always returns one and, with leaf pools that share questions as here, adds no label either; typing saves the end-to-end pass, and labels only for pools that do not share them. The typed certificate issues only where that recalibrated leaf certifies – 41 of 200 draws on the published 3B→7B chain (25 to 54 over five generations), and 0 to 43 on a weaker-upstream chain (Figure 1). **(C5) A worst-case guarantee whose price is measured and whose bite is not observed.** The budget rules out 20 of 64 configurations at any data. Without the correction, none of 48 configurations puts the family-wise error significantly above its target on the test-fold proxy (largest 0.060 against 0.05); in a 3,000-split replication over 64, one does (3/3 certifying splits violate, $p = 1.25 \times 10^{-4}$), in both conditions alike: the correction was never shown to be needed for validity on these tasks. What it does is measurable – certification power and coverage go down – and in the replication the six cells that differ all favour it, the strongest surviving a Holm correction (§7).

2 Why Composition Needs an Account

The per-stage guarantee. Learn-then-Test (Angelopoulos et al., 2025) takes a bounded loss $L(t;\cdot) \in [0,1]$ indexed by a threshold t, tests the null $R(t) = \mathbb{E}[L(t)] > \alpha$ for each t on an a-priori grid with a valid p-value, and reports the thresholds surviving a family-wise correction; any reported t satisfies $R(t) \le \alpha$ with confidence $1 - \delta$, assuming only that calibration and deployment points are i.i.d. from one law. A threshold tuned rather than calibrated states no such bound (Xiao et al., 2025). This *leaf certificate* is the only statistical fact we take as given.

Why chaining leaves is not automatic. Suppose a draft-then-check pipeline feeds a gate $\mathsf{cgen}[\alpha_1]$'s admitted answer to a second gate $\mathsf{cgen}[\alpha_2]$. Both gates search their grids against calibration data; uncounted, the probability that *some* gate certifies a threshold whose true risk exceeds its level is not δ but grows with the number of searches. And even when no gate mis-certifies, the

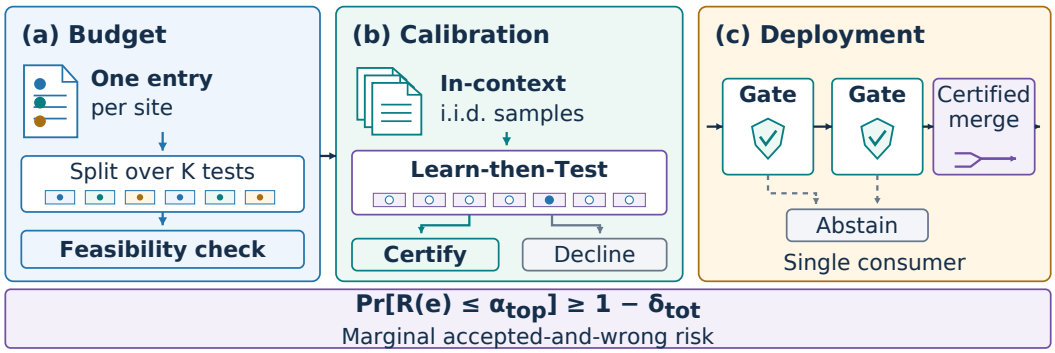


Figure 2: λ-Cert end to end. *(a)* The confidence budget is split over calibration sites, each spreading its δ_i over a K-point grid, which fixes before any data is read the calibration size a binary-loss cell needs, $n_{\text{cal}} \geq \ln(\delta_i/K)/\ln(1-\alpha)$. *(b)* Each site runs one Learn-then-Test search and certifies a leaf or declines, so the static site count is the number of searches run. *(c)* A feed-forward program of gates and a certified aggregator, closed loops outside the certifiable fragment (Theorem 4); typing sums the risk-bearing leaves and Theorem 3 gives $\Pr[R(e) \leq \alpha_{\text{top}}] \geq 1 - \delta_{\text{tot}}$.

output loss must be attributable to some stage: a combiner that manufactures a bad output from acceptable inputs breaks any attempt to read the pipeline's risk off its parts.

λ-Cert answers the first with a resource and the second with a data-flow discipline. The calibration budget is a *linear* context, so the static count of calibration sites equals the number of tests run and a Bonferroni split of the confidence budget is valid even under arbitrary dependence between stages. The data-flow restriction (feed-forward, single-consumer, every merge, whatever fusion it implements (Xu et al., 2026), routed through a certifying aggregator) makes every terminal loss path cross a node that calibrates it – both decidable properties of the program text.

What is certified. We write $L \in \{0,1\}$ for a marker on the deployed output and $R(e) = \mathbb{E}_{\omega\sim P}[L([\![e]\!]_\omega)]$ for the deployment risk, uncomputable because P is unknown. Nothing depends on what L marks; in our experiments it is answer incorrectness, so the certified quantity is the rate of accepted wrong answers over the pool, a marginal loss, and not the selection-conditional error $\Pr[L \mid \mathit{accept}]$, which Table 2 reports beside it as "cond.". A checker that settles the marker outright needs no level and is silent where it does not apply. The certified event is the marginal $\Pr[\mathit{accept} \wedge L] \leq \alpha$, abstention being loss-free, so each certificate is reported with its retained coverage.

3 The λ-Cert Discipline

Judgments. λ-Cert is a call-by-value λ-calculus over an unknown-distribution oracle, a certified-generation gate cgen and a certified aggregator agg. The judgment $\Gamma; \Delta \vdash e : \tau \,!\, \alpha$ pairs an ordinary context Γ with a *linear* calibration budget Δ, a multiset of (sample count, confidence allowance) pairs, and a risk effect α (Figure 6). The absence of contraction on Δ bounds the number of calibration tests statically, which is exactly what makes Theorem 3's Bonferroni allocation sound.

Well-formedness. Three decidable side conditions delimit the feed-forward fragment, and together they make the ledger's static site count equal the runtime test count: no gate's output reaches its own input, checked by forbidding gates in recursive positions over an acyclic let-graph; every gate-derived value has a single consumer, two such meeting only at an aggregator; and a function closing over unspent budget is used affinely. The first is structural, the syntax having no fixpoint form, and the third is subsumed in this first-order fragment; single-consumption is the one needing a check, the decidable predicate wf of the mechanization.

What typing guarantees, and what it assumes. This precision is what makes "certify by typing" a claim rather than a slogan, so Table 1 states it as a list. The three declared obligations are declared in the way Fuzz (Reed & Pierce, 2010) declares primitive sensitivities: they quantify over an uncomputable loss predicate against an unknown distribution, which no type system can decide.

Table 1: The split "certify by typing" depends on. Left: decided by the checker and mechanized. Right: stated by the judgment, discharged by calibration at deployment.

machine-decidable, enforced by typing	declared, discharged at deployment
feed-forward: no gate reads its own output	the per-stage Learn-then-Test certificate
single-consumer: no fan-out off a gate	loss-monotone composition
affine budget use; the linear δ-ledger; effect soundness (Lemma 1)	a loss-free abstain fallback, in context

4 THE COMPOSITION THEOREM

Two union bounds. Soundness rests on two accountings. *Confidence:* the linear budget bounds the number of tests statically, so a Bonferroni split $\sum_i \delta_i \le \delta_{\mathsf{tot}}$ controls the probability that any stage mis-certifies, under arbitrary dependence. *Risk:* on the event that none does, output loss is charged to some risk-bearing leaf, and the effect sums those charges.

Lemma 1 (Effect-soundness, mechanized) *If $\Gamma; \Delta \vdash e : \tau \,!\, \alpha$ then α is the sum of the effects of all risk-bearing leaves of e: oracles at 1, gates and aggregators at their level. No loss source escapes α_{top}.*

Lemma 2 (Loss decomposition) *For a well-formed feed-forward program, on the event that no stage mis-certifies, any output loss traces to a single risk-bearing leaf, so $R(e) \le \sum_i \pi_i \alpha_i \le \alpha_{\mathsf{top}}$ with π_i the probability that leaf i is reached.*

Theorem 3 (Composed risk certificate) *Let $\vdash e : \tau \,!\, \alpha_{\mathsf{top}}$ with budget δ_{tot}, where e is well-formed (FF, SC, AFF), every gate uses an abstain fallback, losses lie in $[0,1]$, and each stage is calibrated on an i.i.d. in-context sample of the inputs reaching it against an a-priori grid. Then $\Pr[\, R(e) \le \alpha_{\mathsf{top}} \,] \ge 1 - \delta_{\mathsf{tot}}$.*

The proof is in Appendix B (Figure 7); §5 states the part Rocq checks. The premise doing the most work in practice is *in-context* calibration: stage i must be calibrated on the inputs reaching it under the deployed upstream policy. Because the fragment is feed-forward, that distribution is well defined and computable in data-flow order rather than a fixpoint of stage i's own threshold – the property Theorem 4 shows closed loops lack.

Gating, aggregation and inference. Gate insertion is *risk-non-increasing*: $\mathsf{cgen}[\alpha](e, \mathsf{abstain})$ is well-typed at effect α, agrees with e on the accept sub-event, and returns the loss-free fallback otherwise. Aggregation is what lets λ-Cert cover DAGs: rather than forbid combiners that manufacture loss from acceptable inputs, an aggregator calibrates the composite loss $L(\phi(\cdot))$ *directly*, needing no subadditivity assumption (Appendix B). Inference is one syntax-directed traversal $\mathsf{infer}(e) = (\mathrm{hsum}(e), \mathrm{dsum}(e))$, total, sound and complete, so type checking is decidable and the deployer reads off the α_{top} and δ_{tot} a deployment must supply.

No static certificate for closed loops. The feed-forward restriction is not an artifact of the proof technique: a closed-loop agent reads its own past outputs, so a stage's deployment law $P = \Phi(\hat{t})$ depends on the threshold being chosen, and a sample drawn before deployment need not say which.

Theorem 4 (Impossibility) *Fix $\alpha \in (0,1)$ and a utility floor $\rho > 0$. Let Φ_0, Φ_1 be closed-loop dynamics agreeing at the pre-deployment policy, so a calibration sample S has one law under both, and* adversarially placed*: at every threshold accepting mass at least ρ, the deployed risk exceeds α in at least one world. Then every static calibrator whose selected threshold always accepts mass at least ρ – selecting $\hat{t}$ as a measurable function of S, with or without independent randomization – has $\sup_b \Pr[R_{P^b}(\hat{t}) > \alpha] \ge \frac{1}{2}$ at every sample size.*

Both hypotheses are properties a closed loop can have: dynamics a pre-deployment sample cannot tell apart, and a deployment law that places the loss on the accepted side of whichever threshold is deployed. This is the scoped analogue of impossibility results in performative prediction (Perdomo et al., 2020), and points at on-policy or iterative recalibration under a contractivity assumption as the route to certifying restricted loops. Appendix E proves it and exhibits a pair: two dynamics, each certifiable on its own under the floor, that no pre-deployment sample tells apart (Proposition 5).

5 MECHANIZATION

The development is nine Rocq modules behind a single `make`. *Checked with no axioms:* a finite-probability union bound $\Pr[\bigcup_i \mathrm{BAD}_i] \le \sum_i \delta_i$, and the capstone soundness_full_fused, which in one proof applies the deterministic core to the typing derivation and feeds the good-event bound to the union bound: if $\vdash e$ at an $\mathbb{N}$-valued level a, $\Pr[\mathrm{BAD}_i] \le \delta_i$ for each certificate and the nine per-node bounds hold on the good event, then $\Pr[R(e) \le a] \ge 1 - \sum_i \delta_i$. `Print Assumptions` on both reports *Closed under the global context*. *Checked modulo the loss semantics:* effect-soundness, budget conservation, the good-event bound and decidable principal inference, the semantics *deriving* six of the nine per-node obligations. The remaining assumption surface is four `Parameters` – the world space and the oracle, gate and aggregator predicates – on which only the semantics-level theorems depend. The accounting is re-proved over $\mathbb{Q}$, with a worked pipeline at $0.02{+}0.03 = 0.05$. *On paper only:* the cited Learn-then-Test certificate; the budget step $\sum_i \delta_i \le \delta_{\mathsf{tot}}$, since Rocq does not tie the δ_i to the typed budget, and with it Theorem 3 as stated; and Theorem 4 (Appendix E), since closed loops are not expressible in the acyclic-by-construction syntax the development checks.

6 EXPERIMENTS

Setup. We instantiate a gate by *selective self-consistency* (Wang et al., 2023): the generator samples $k{=}8$ chains per question, the deployed answer is the majority vote, the score is its vote share, and the loss marker L is that answer being wrong under exact match. We use the Instruct models of Qwen2.5 at 1.5B, 3B, 7B and 14B, Llama-3.1-8B and OLMo-2-7B over GSM8K (Cobbe et al., 2021) and ARC-Challenge (Clark et al., 2018); because the construction is stated over an abstract bounded loss marker, we also run it with the marker produced by a different modality, Qwen2.5-VL-Instruct at 3B and 7B over the image-bearing questions of ScienceQA (Lu et al., 2022) and A-OKVQA (Schwenk et al., 2022), both multiple choice and run by the *same* code as ARC. Each cell uses $n{=}1000$ questions over 200 random calibration/test splits at $\alpha{=}\delta{=}0.05$ with an a-priori grid of 51 thresholds; within a task all models answer the same question set, so gates from different models share calibration items.

RQ1: do per-stage certificates hold on real models? Table 2 says they do on GSM8K at all four scales, empirical guarantee coverage at or above the nominal $1 - \delta = 0.95$. Retained coverage rises with model quality at a fixed certified level (0.36 at 1.5B to 0.92 at 14B); all four scales calibrate at $\hat{t} \approx 0.76$. ARC-Challenge marks the boundary: a wrong but consistently selected option still draws a high vote share, so unanimity over all eight chains is wrong on 48 of 884 questions at 7B against 26 of 676 at 3B (two-sided Fisher $p{=}0.15$). At $\alpha{=}0.05$ it *declines* to certify the 1.5B and 7B gates, declines 14B in Table 2's file order (in question-id order it certifies on a few draws per generation, Table 10), and certifies the better-separated 3B gate on only 42 of 200 calibration draws, none of which holds α out of sample: the fourth scale adds a point to the GSM8K curve and a decline on ARC. The bottom block of Table 2 repeats the construction with the answer read off an image, changing nothing in gate, calibrator or marker, and the pattern reappears. ScienceQA certifies at both scales, on 54 of 200 draws at 3B and 141 at 7B, with median coverage rising from 0.55 to 0.79, and A-OKVQA is the boundary case ARC is for text. Of the seven declines in Table 2, four share one cause (§7; two in Figure 3), 1.5B/ARC has no admissible threshold, and the two added families' ARC cells, the highest base error rates, decline outright. **Certificates hold wherever the vote share separates – GSM8K at four scales, ScienceQA at two – and are refused where it does not, none of them met by abstaining.**

Does Table 2 survive two further generation runs, and a change of family? Fourteen of its sixteen cells keep one verdict, read in question-id order over the shared thousand (Table 10), and the two that do not sit on the boundary. Every cell was re-generated twice under a recorded seed each time, and for six Qwen text cells a newer vLLM (Appendix D). Both ends reproduce exactly: the five GSM8K cells certify on all 200 draws in every generation (14B on its full split; 199 on the shared pool), the four that never certify never do, and each added family lands where Qwen2.5-7B does on its dataset. The exception is VL-7B/A-OKVQA, which certifies in two of the three, on 0, 7 and 3 of 200 draws: one of the four cells sitting on the boundary; the other non-reproducing cell, Llama-8B/ARC, certifies in one generation. Two further generations on a second machine keep 8 of 10 Qwen cells unanimous over five; VL-7B/A-OKVQA (2/5) is read as unstable (Appendix D).

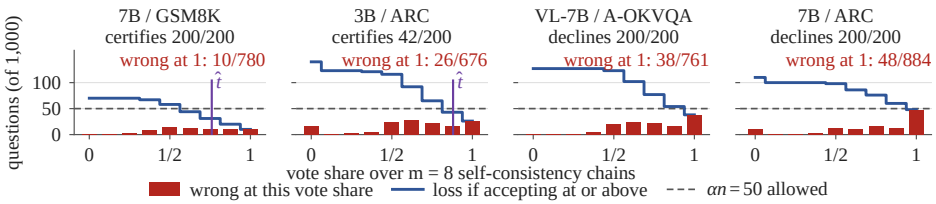


Figure 3: Where the loss falls, and why two of these four gates decline. Bars: wrong answers at each attainable vote share over m=8 chains. Step: the loss a gate accepting at or above that share would carry over the 1,000-question pool, right to left. All four sit under $\alpha n = 50$ at unanimity, so a point-estimate rule ships all four; certification is decided on 700-question folds at δ/K (§7).

Table 2: RQ1. Per-stage certificates at (α, δ)=$(0.05, 0.05)$. The certified quantity is the marginal $\Pr[accept \wedge L] \leq \alpha$; "coverage" is median retained acceptance, "cond." is $\Pr[L \mid accept]$, "emp. guar." the fraction of *certified* splits whose out-of-sample risk respected α: selection-conditional, consistent with soundness at $\geq 1-\delta = 0.95$ but no test of it (0.975 is $5/200$, exact 95% upper bound 0.052). "–": declined. † certified on $42/200$ (3B/ARC), $54/200$ (VL-3B/SciQA) and $141/200$ (VL-7B/SciQA) draws (file order, as in Figure 3 and §7; the replication uses qid order, Appendix D); these are also the denominators, so no violation in 42 or 54 draws bounds the rate only at 0.069 or 0.054, above δ, while none in 141 gives 0.021, below it. ‡ unanimity error on α in these four cells; Figure 4 gives every cell's verdict and certifying count over the three generations. Bottom four rows read the answer off an image.

model	task	base err.	$\hat{t}$	coverage	cond.	emp. guar.
Qwen2.5-1.5B	GSM8K	0.223	0.76	0.36	0.033	0.975
Qwen2.5-3B	GSM8K	0.112	0.76	0.74	0.023	1.000
Qwen2.5-7B	GSM8K	0.070	0.76	0.86	0.023	0.995
Qwen2.5-14B	GSM8K	0.049	0.76	0.92	0.024	1.000
Qwen2.5-1.5B	ARC	0.266	–	–	–	–
Qwen2.5-3B	ARC‡	0.140	0.88	0.67†	0.038	1.000†
Qwen2.5-7B	ARC‡	0.110	–	–	–	–
Qwen2.5-14B	ARC‡	0.073	–	–	–	–
Llama-3.1-8B	GSM8K	0.116	0.76	0.70	0.025	1.000
OLMo-2-7B	GSM8K	0.116	0.88	0.61	0.025	1.000
Llama-3.1-8B	ARC	0.195	–	–	–	–
OLMo-2-7B	ARC	0.250	–	–	–	–
Qwen2.5-VL-3B	ScienceQA	0.202	0.88	0.55†	0.047	1.000†
Qwen2.5-VL-7B	ScienceQA	0.095	0.88	0.79†	0.028	0.993†
Qwen2.5-VL-3B	A-OKVQA	0.162	–	–	–	–
Qwen2.5-VL-7B	A-OKVQA‡	0.127	–	–	–	–

What does *not* reproduce is how often a certificate is issued (Figure 4): the widest cell spans 149 draws of 200, and four of the sixteen triples fall and then rise or the reverse, so no pair of generations reads as a move. Marginal guarantee coverage stays at 0.980 or above in all 96 arms: no guarantee is missed, only the rate is unstable. We therefore read Table 2's $k/200$ counts as single draws rather than properties of a cell, and its four ‡ rows as not distinguishable at n=1,000 rather than certify/decline facts; RQ2–RQ3 use the GSM8K leaves only, and every RQ4 cell drawing on an ARC or A-OKVQA leaf also draws on one that declines in all three. **The verdict reproduces in** 14 **of** 16 **cells over** 3 **generations; the certifying rate does not (up to** 149 **draws of** 200**).**

RQ2: does typed composition survive a genuine dependent pipeline? Table 3 runs a parallel pair and two dependent chains on real data, the chain's downstream gate calibrated in-context on the upstream gate's *actually-accepted* outputs. The pair answers only when both gates accept, is typed at $\alpha_{\mathsf{top}} = 0.10$ and certifies on all 200 splits at a median composed risk of 0.0067: pairing does not shrink either gate's calibration pool. The aggregation rule, which calibrates the combined loss, is not run on real models. The 3B→7B chain certifies on 41 of 200 splits and respects α_{top} on every one; the 1.5B→3B chain declines on all of them, for the reason §7 reads off its calibration

Table 3: RQ2. Composition on real models (GSM8K, 200 splits, $\alpha{=}\delta{=}0.05$ per leaf, so $\alpha_{\mathsf{top}}{=}\delta_{\mathsf{tot}}{=}0.10$). "Cert." counts splits where both leaves certify, a chain's downstream leaf recalibrated on the upstream-accepted subset; "realized" the median out-of-sample composed risk there, on the dependent (upstream-accepted) subset with the downstream gate recalibrated per split, a different object from Table 11's fixed-threshold pipeline.

real pipeline	cert.	median realized	holds $\leq \alpha_{\mathsf{top}}$
parallel pair, both accept (1.5B + 7B)	$200/200$	0.0067	100%
dependent chain (3B $\rightarrow$ 7B)	$41/200$	0.0152	100%
dependent chain (1.5B $\rightarrow$ 3B)	$0/200$	declines	–

mass. **The pair certifies on every split far under α_{top}; a dependent chain certifies only where the upstream leaves enough calibration mass behind it.**

RQ3: what does composing by typing cost against certifying end to end? The alternative to reading a pipeline's guarantee off its typing is an exact binomial bound on end-to-end labelled runs. Table 12 puts both on one object: the 3B$\rightarrow$7B pipeline, both thresholds fixed at the certified 0.76 of Table 2, a loss if either stage errs, the evaluation fold of each of 200 splits, coverage judged on the fold neither bound read. Realized risk is 0.010; the typed bound $\alpha_{\mathsf{top}} = 0.10$ holds on all 200 splits; a Clopper–Pearson bound on the fold's own 300 end-to-end labels at the same δ_{tot} is 0.022 and holds on 0.950 of them: the tighter certificate, computed from the labels, while the typed one sums leaf levels fixed at 0.05 in advance; we ran no sweep over leaf levels. Resampling those labels with replacement tightens the printed number and drops the guarantee: coverage falls from 0.894 at 200 draws to 0.670 at 2,000, so those rows are measurements. On labels the two tie, every row drawing none, since both leaf pools sit on the same questions. What typing buys is the pass: leaf certificates (700 labels each) serve every well-typed program over the same leaves; an end-to-end bound is collected per pipeline. On availability the order reverses: the end-to-end bound always returns, the typed certificate only where the recalibrated downstream leaf certifies on the upstream-accepted subset – $41/200$ draws here (25–54 over five generations, Table 8), $0/200$ on 1.5B$\rightarrow$3B. **Typing needs no end-to-end pass, saves no label here, states a bound 4.5 times looser, and certifies only where the recalibrated downstream leaf does.**

RQ4: when does the calibration-budget accounting actually bind? The linear budget exists to stop unaccounted reuse from inflating the family-wise error. Sweeping the grid size from 5 to 201 over six values of K (Figure 5a), the corrected certificate posts a flat 0.015 violation rate at every K; the uncorrected one matches it at $K{=}5$ and posts 0.020 from $K{=}11$ up, which at $K{=}51$ is 3 against 4 violations in the same 200 paired splits. The correction's visible effect is on coverage, which it costs 10.7 points at $K{=}5$ and 3.7 to 4.7 points above that. Over one to three stages sharing calibration items the two post *identical* family-wise error at every k (the correction moves the threshold on 22 and 49 of 200 splits at $k{=}2, 3$). **On these tasks the correction is never needed for validity on the test-fold proxy: no cell of the 48 puts the uncorrected family-wise error significantly above its target (largest 0.060 against 0.05, Appendix D), the one replication cell that does exceeds it in both conditions, and 3 against 4 violations in 200 splits resolves nothing. Its measurable price is coverage; the 3,000-split replication resolves the direction of the difference (§7).**

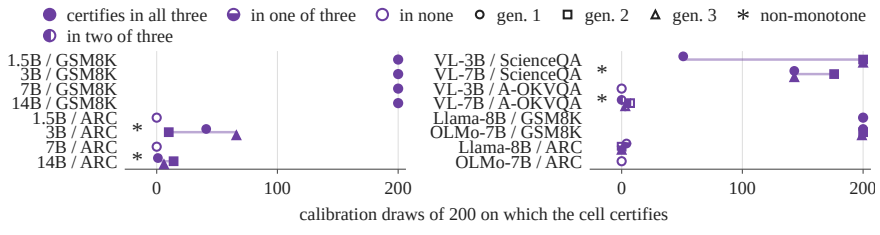


Figure 4: The verdict reproduces across generations; the certifying rate does not. One row per cell of Table 2, continued in the right panel; a single point where the three generations agree, else circle, square, triangle for generations 1–3 over the row's span; colour is the verdict (hollow: none). Starred rows are the four whose triple falls then rises or the reverse, so no two generations read as a move. All arms certify in question-id order (Table 2†).

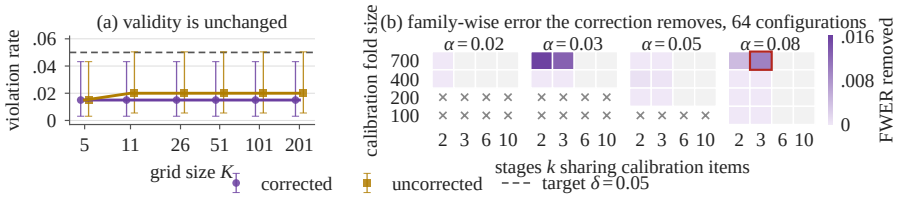


Figure 5: The budget correction against its absence. *(a)* Violation rate on the 7B GSM8K gate against grid size, with exact Clopper–Pearson intervals on the integer counts: at every $K > 5$ the gap is one split of 200 (none at K=5; Table 13 has these rows and the median coverage, exactly). *(b)* The 3,000-split sweep; shade is family-wise error the correction removes, 6 of 64 cells differing in it, corrected lower in every one. Grey: no split certifies; ×: none could, ruling out 20 before any data; red: survives Holm.

7 Analysis: What Refuses a Certificate

What refuses a certificate. Figure 3 puts four decisive cells side by side: scoring over m=8 chains places most of every pool at unanimity, and with it most of the loss a gate would accept. Four of the five Qwen2.5 and Qwen2.5-VL declines have one cause (the two added families' ARC cells, the two highest base error rates, are not analysed here). Unanimity across all eight chains is wrong on $42/664$ items at VL-3B/A-OKVQA, $38/761$ at VL-7B/A-OKVQA, $48/884$ at 7B/ARC and $36/915$ at 14B/ARC: 0.063, 0.0499, 0.054 and 0.039 *given* acceptance, but 0.042, 0.038, 0.048 and 0.036 of the pool, and it is that second, marginal reading the certificate bounds. All four sit under α there, so what refuses them is not the point estimate but the finite-sample correction at δ/K on a fold of 700: a rule thresholding the *empirical* error would ship all four, and a later generation moves one across the line. 143 of 7B/ARC's 200 declines, 200 of 14B/ARC's 200, 200 of VL-7B/A-OKVQA's and 198 of VL-3B/A-OKVQA's refused an admissible point estimate, while 1.5B/ARC refused 0: no grid threshold there is admissible at all. Appendix D prices both candidates: of the declining cells the K=9 grid moves only VL-7B/A-OKVQA, in part; finer vote counts rescue 3B/ARC alone. **What refuses a certificate is the distance from α measured in the fold's own noise, not the point estimate; the cells nearest that boundary are the ones whose certifying count moves between generations (RQ1).**

Two predictions, and the sweep that tests them. Certification needs a threshold whose Hoeffding–Bentkus p-value is at most δ_i/K, and the least any fold can attain is $(1-\alpha)^{n_{\text{cal}}}$, so a cell certifies only if $n_{\text{cal}} \geq \ln(\delta_i/K)/\ln(1-\alpha)$ (Figure 2, bottom; Appendix D). That reads off the configuration alone and rules out 20 of 64 configurations at any data (Figure 8), by arithmetic, not measurement. The bound is necessary, not sufficient: 24 cells it allows also certify on no split (a declining leaf in their stage set), so the counts below are against the 20 cells that *do* certify. The second prediction is mechanistic: Theorem 3 is worst-case, so the union-bound cost should move the family-wise error only where the margin from the calibrated risk to α approaches that estimate's finite-sample fluctuation. We swept 48 configurations of level, fold size and stage count into the thin-margin regime, 300 splits per cell. Thirty-one cells certify on no split in either condition; of the seventeen that do, six certify on different numbers of splits – the certification power the correction costs, as steep as 19 against 83 – and four post a different family-wise error; the two conditions differ on eight cells in one dimension or the other and on two in both. Under the criterion the sweep encodes – uncorrected above δ where corrected is not – exactly one separates them: $\alpha = 0.03$, fold 700, $k = 3$, uncorrected 0.060 against corrected 0.000. We decline to claim it: those are 5 violations in 83 certified splits against 0 in 19, and a two-sided 95% Clopper–Pearson interval on $5/83$, $[0.020, 0.135]$, contains the 0.050 target. The other three keep both conditions under δ (0.043 against 0.034, 0.017 against 0.003, 0.027 against 0.010), corrected lower in all four – the direction the theory predicts – and none of them resolvable at 300 splits (smallest two-sided Fisher $p = 0.22$). No cell of the 48 puts the uncorrected error significantly above δ: the closest, that $5/83$, gives a one-sided exact $p = 0.40$. A replication at 3,000 splits per cell, with the four vision–language leaves appended to the pool set (64 cells), resolves both: the borderline cell drops to 28 violations in 809 certified splits, 0.035, no cell of the 20 that certify satisfies the biting criterion, and the direction resolves – corrected lower in all six differing cells, the strongest at two-sided Fisher $p = 0.0014$, surviving a Holm correction over the six. One count goes the opposite way: at $\alpha = 0.02$ all 3 certifying folds violate ($p = 1.25 \times 10^{-4}$ against δ), in both conditions alike, marginally $3/3{,}000$.

This is the predicted pattern: with a wide margin ($\alpha = 0.05$ against base error 0.07 on a fold of 700) the correction moves the family-wise error not at all, and every cell where it moves has a margin the fold can barely resolve. **The correction moves the family-wise error only where the margin is narrow.**

Negative results, and a chain at the edge. Two practices, claiming the terminal level or $\max_i \alpha_i$ rather than the sum, go unviolated across 200 splits (0.000, 0.000) yet bound nothing; Šidák and Holm leave the family-wise error at Bonferroni's 0.020 at $k{=}3$ and median coverage within 0.22 points. The weaker-upstream chain, 1.5B$\rightarrow$3B, certifies 0 to 43 of 200 splits across 5 generations (0 on the original): the weak gate passes 111.6 to 204.4 questions against 223.7, and the downstream error on that subset (0.0267–0.0344) sits at the edge of what the fold can certify. **Where the upstream is weak, the certificate is a property of the draw rather than of the chain.**

8 Related Work

Distribution-free risk control. Our certificate descends from conformal prediction (Vovk et al., 2022; Angelopoulos & Bates, 2022); bounded risk originates in risk-controlling prediction sets (Bates et al., 2021), generalized by conformal risk control (Angelopoulos et al., 2024) and Learn-then-Test (Angelopoulos et al., 2025), all for a *single* predictor.

Selective prediction and certified gating for LLM systems. Abstention operationalizes a risk budget, from the error–reject trade-off (Chow, 1970; El-Yaniv & Wiener, 2010) to calibrated selection (Geifman & El-Yaniv, 2017; Fisch et al., 2022); conformal and risk-control guarantees for LLM outputs (Quach et al., 2024; Mohri & Hashimoto, 2024; Cherian et al., 2024; Kang et al., 2024; Wang et al., 2024; Xiao et al., 2026b) sit at *one* decision point.

System-level and compositional certification. External sampling certifies a fixed system under an end-to-end binomial bound, re-run per system (RQ3's baseline). Pipeline-aware conformal prediction (Kotte, 2026a) gives joint coverage across stages for set-valued outputs, role-stratified risk control (Rahman et al., 2026) splits one budget across tool-call roles, post-hoc trajectory-risk certification (Li, 2026) bounds modular agents after the fact, and a map of when conformal risk control can certify structured generation at all (Kotte, 2026b) sets the scope; none types the calibration budget. The closest prior *compositional* result, conformal prediction for neurosymbolic programs (Ramalingam et al., 2024), propagates *prediction sets* by abstract interpretation where we certify a *risk level* of an abstaining pipeline; a set-valued output has no accept event, so no head-to-head exists.

Effect systems, graded resources and mechanized guarantees. The machinery is the type-and-effect tradition (Gifford & Lucassen, 1986; Lucassen & Gifford, 1988), graded monads (Katsumata, 2014) and coeffect calculi (Petricek et al., 2014; Brunel et al., 2014; Gaboardi et al., 2016; Orchard et al., 2019). Sensitivity and privacy type systems (Reed & Pierce, 2010; Gaboardi et al., 2013; Barthe et al., 2016; Aguirre et al., 2024) and typing what an LLM may emit (Xiao et al., 2026a) make typing entail a bound; ours is such a grade but indexes held-out samples whose non-reuse keeps the exchangeability conformal bounds need, a failure no usage grading we know of expresses.

9 Limitations and Conclusion

The additive $\alpha_{\mathsf{top}} = \sum_i \alpha_i$ degrades with depth and certifies nothing once the sum reaches one. In-context calibration costs labels at each stage; RQ2's weak chain, short of them, certifies on 0 to 43 of 200 splits over 5 generations. We certify answer error, not harmfulness. A gate on α is unresolved at $n{=}1{,}000$: fourteen of Table 2's sixteen verdicts hold over three generations (not 14B/ARC or VL-7B/A-OKVQA) while certifying rates span up to 149 of 200 draws (RQ1).

Typing composes per-stage certificates into $\Pr[R(e) \le \alpha_{\mathsf{top}}] \ge 1 - \delta_{\mathsf{tot}}$ without end-to-end recalibration, and on real models certifies a dependent chain only where the recalibrated downstream leaf does. In the 64-cell replication the budget posts fewer violations in all six differing cells and one $\alpha{=}0.02$ cell violates in both (§7). Closed loops lie outside it (Theorem 4).

AI USE STATEMENT

Generative AI tools were used substantially in the preparation of this work, and we disclose the uses that fall under the required categories. LLMs helped develop the conceptual framework, propose and refine hypotheses about which fragment of the design space was provable, formulate mathematical claims, assist in the writing of proofs, and give feedback on research methodology and experiment design, including an adversarial review pass over an earlier draft whose findings shaped the present evaluation. They also implemented methods, namely the prototype type checker and calibrator, the Rocq development and the evaluation harness, and assisted in interpreting results. Under the recommended categories, they were additionally used for code creation, literature search and analysis, brainstorming, and editing the text of this paper.

We have *not* used generative AI tools to produce, select or filter any experimental result: every number in this paper is emitted by the scripts in the artifact from cached generation records, and the research questions, the claims and the conclusions are the authors'. The remaining required-disclosure categories are not applicable to this work, which has no human-subject component and no data collection beyond public benchmarks and the cached outputs of open-weight models.

We have reviewed all AI-assisted work and take responsibility for the final content, including every claim, number and artifact. Two properties of this paper make that review checkable rather than asserted. The composition step is machine-checked: a proof assistant, not an author or a language model, verifies the Rocq statement of §5 (soundness_full_fused, at $\mathbb{N}$-valued levels, with conclusion $1 - \sum_i \delta_i$) and reports its assumption set. Theorem 3's budget step $\sum_i \delta_i \leq \delta_{\mathsf{tot}}$, the cited leaf certificate and Theorem 4 are proved on paper only. Every empirical number replays on CPU from cached generation runs shipped with the artifact. The claims we could not support this way we removed: an earlier version of this work asserted that the calibration-budget discipline prevents violations, and RQ4 reports the measurements that contradicted it.

REPRODUCIBILITY STATEMENT

The supplementary archive (`mech/`, `prototype/`, `eval/`) contains the Rocq development, which reproduces the machine-checked metatheory of §5 under a single `make` together with the `Print Assumptions` transcript; the prototype type checker and Learn-then-Test calibrator; the cached per-question generation records for all sixteen model/task cells of Table 2, with their further generations (Table 10); and the scripts producing every table in §6. All evaluation runs on CPU from the cached records with fixed split seeds, needing no GPU and no network; the generation runs behind the cache were performed once per cell and unseeded, except that the cells of Table 2 were re-generated under seeds 20260828 and 20260829, with further generations on a second machine (one under a batch cap of 16 on an RTX 3090; Appendix D lists every generation, its machine and its stack), all shipped beside their originals as the replications reported in §6. The records were not all produced on one inference stack: the original Qwen2.5 GSM8K and ARC cells were generated under `vllm==0.6.3.post1`; the vision–language cells and the second and third generations of the 1.5B–7B cells under `vllm==0.7.3` with `torch==2.5.1+cu121` and `transformers==4.57.6`; the Llama-3.1 and OLMo-2 cells and generations four and five under `vllm==0.7.3` with `transformers==4.49.0`; and the 14B cells under `vllm==0.7.3` (`eval/runs/ENGINE_PROVENANCE*.txt` and `eval/READOUT_B2_14B.md` give each record's stack). An engine banner in a generation log attests the stack only for generations two to five of the ten 1.5B–7B Qwen2.5 and Qwen2.5-VL cells. The stacks named above for the original generation of those cells, for the Llama-3.1 and OLMo-2 cells and for the 14B cells are not attested: no generation log backs them, and they rest on the authors' run notes and the pre-registration. Both pins ship with the artifact, and Appendix D states where that difference qualifies a comparison. Assumptions of Theorem 3 are listed in its statement and §3; full proofs are in Appendix B.

ETHICS STATEMENT

This work certifies a bounded loss on a pipeline's output and instantiates that loss as answer error on two public reasoning benchmarks. We flag one misreading worth avoiding: a certificate of this kind bounds a rate under an i.i.d. in-context calibration assumption, and it neither detects nor bounds harm in any sense beyond the marker it is calibrated against. Deploying such a certificate as a safety

control requires a loss marker that genuinely tracks the harm of concern and a calibration sample drawn from the deployment distribution; neither is supplied by the type system, and §9 states that we do not evaluate harmfulness.

REFERENCES

Alejandro Aguirre, Philipp G. Haselwarter, Markus de Medeiros, Kwing Hei Li, Simon Oddershede Gregersen, Joseph Tassarotti, and Lars Birkedal. Error credits: Resourceful reasoning about error bounds for higher-order probabilistic programs. *PACMPL*, 8(ICFP), 2024. doi: 10.1145/3674635.

Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv:2107.07511, 2022.

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *ICLR*, 2024.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *Annals of Applied Statistics*, 19(2), 2025. doi: 10.1214/24-AOAS1998.

Gilles Barthe, Marco Gaboardi, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. A program logic for union bounds. In *ICALP*, 2016. doi: 10.4230/LIPIcs.ICALP.2016.107.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, et al. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 2021. doi: 10.1145/3478535. arXiv:2101.02703.

Aloïs Brunel, Marco Gaboardi, Damiano Mazza, and Steve Zdancewic. A core quantitative coeffect calculus. In *Proceedings of the 23rd European Symposium on Programming (ESOP 2014)*, 2014. doi: 10.1007/978-3-642-54833-8_19.

John J. Cherian, Isaac Gibbs, and Emmanuel J. Candès. Large language model validity via enhanced conformal prediction methods. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024. arXiv:2406.09714.

C. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 1970. doi: 10.1109/TIT.1970.1054406.

Peter Clark, Isaac Cowhey, Oren Etzioni, et al. Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv:1803.05457, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. arXiv:2110.14168, 2021.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 2010.

Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Calibrated selective classification. arXiv:2208.12084, 2022.

Marco Gaboardi, Andreas Haeberlen, Justin Hsu, Arjun Narayan, and Benjamin C. Pierce. Linear dependent types for differential privacy. In *POPL*, 2013. doi: 10.1145/2429069.2429113.

Marco Gaboardi, Shin-ya Katsumata, Dominic A. Orchard, et al. Combining effects and coeffects via grading. In *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming (ICFP 2016)*, 2016. doi: 10.1145/2951913.2951939.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017. arXiv:1705.08500.

D. Gifford and J. Lucassen. Integrating functional and imperative programming. In *Proceedings of the 1986 ACM Conference on LISP and Functional Programming (LFP '86)*, 1986. doi: 10.1145/319838.319848.

Mintong Kang, Nezihe Merve Gürel, and Ning Yu. C-rag: Certified generation risks for retrieval-augmented language models. In *International Conference on Machine Learning (ICML 2024)*, 2024. arXiv:2402.03181.

Shin-ya Katsumata. Parametric effect monads and semantics of effect systems. In *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2014)*, 2014. doi: 10.1145/2535838.2535846.

Varun Kotte. Pasc: Pipeline-aware conformal prediction with joint coverage guarantees for multi-stage nlp and llm pipelines. arXiv:2605.18812, 2026a.

Varun Kotte. When can conformal risk control certify llm outputs? bounds, impossibility, and adaptation for structured generation. arXiv:2606.29054, 2026b.

Zhenpeng Li. Post-hoc trajectory-risk certification for modular llm-based security agents. arXiv:2608.05199, 2026.

Pan Lu, Swaroop Mishra, Tanglin Xia, et al. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, 2022.

J. M. Lucassen and D. K. Gifford. Polymorphic effect systems. In *Proceedings of the 15th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 1988)*, 1988. doi: 10.1145/73560.73564.

Christopher Mohri and Tatsunori Hashimoto. Language models with conformal factuality guarantees. In *International Conference on Machine Learning (ICML 2024)*, 2024. arXiv:2402.10978.

Dominic Orchard, Vilem-Benjamin Liepelt, and Harley Eades III. Quantitative program reasoning with graded modal types. In *ICFP*, 2019. doi: 10.1145/3341714.

Juan C. Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, et al. Performative prediction. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, 2020. arXiv:2002.06673.

Tomas Petricek, Dominic Orchard, and Alan Mycroft. Coeffects: A calculus of context-dependent computation. In *Proceedings of the 19th ACM SIGPLAN International Conference on Functional Programming (ICFP 2014)*, 2014. doi: 10.1145/2628136.2628160.

Victor Quach, Adam Fisch, and Tal Schuster. Conformal language modeling. In *International Conference on Learning Representations (ICLR 2024)*, 2024. arXiv:2306.10193.

Md Ashikur Rahman, Md Arifur Rahman, Niamul Hassan Samin, Khandaker Rifah Tasnia, Md Hasibul Amin, Sifat Rahman Ahona, and Juena Ahmed Noshin. Beyond aggregate risk: Role-stratified conformal risk control for llm tool calls. arXiv:2607.24343, 2026.

Ramya Ramalingam, Sangdon Park, and Osbert Bastani. Uncertainty quantification for neurosymbolic programs via compositional conformal prediction. arXiv:2405.15912, 2024.

Jason Reed and Benjamin C. Pierce. Distance makes the types grow stronger: A calculus for differential privacy. In *ICFP*, 2010. doi: 10.1145/1863543.1863568.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, et al. A-OKVQA: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision*, 2022.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2 edition, 2022. doi: 10.1007/978-3-031-06649-8.

Xuezhi Wang, Jason Wei, Dale Schuurmans, et al. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR 2023)*, 2023. arXiv:2203.11171.

Zhiyuan Wang, Jinhao Duan, Lu Cheng, et al. Conu: Conformal uncertainty in large language models with correctness coverage guarantees. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024. arXiv:2407.00499.

$$\frac{}{\Gamma;\cdot \vdash \mathsf{oracle}(e) : \mathsf{O}\,!\,1}\ (\textsc{Oracle}) \qquad \frac{\Gamma;\Delta_1 \vdash e_1 : \tau_1\,!\,\alpha_1 \qquad \Gamma, x{:}\tau_1;\Delta_2 \vdash e_2 : \tau_2\,!\,\alpha_2}{\Gamma;\Delta_1 \uplus \Delta_2 \vdash \mathsf{let}\, x{=}e_1\, \mathsf{in}\, e_2 : \tau_2\,!\,(\alpha_1{+}\alpha_2)}\ (\textsc{Let})$$

$$\frac{\Gamma;\Delta_g \vdash e_{\mathrm{gen}} : \mathsf{O}\,!\,\alpha_g \qquad \Gamma;\Delta_f \vdash e_{\mathrm{fb}} : \tau\,!\,0 \qquad \mathsf{LTT}_Z(\Lambda,\alpha,\delta_c)\ \text{valid}}{\Gamma;\Delta_g \uplus \Delta_f \uplus (\,n@\delta_c) \vdash \mathsf{cgen}[\alpha](e_{\mathrm{gen}}, e_{\mathrm{fb}}) : \tau\,!\,\alpha}\ (\textsc{CGen})$$

$$\frac{(\Gamma;\Delta_i \vdash e_i : \tau_i\,!\,\alpha_i)_{i=1}^{m} \qquad \mathsf{LTT}_Z(\Lambda_\phi,\alpha,\delta_\phi)\ \text{valid}}{\Gamma;(\biguplus_i \Delta_i) \uplus (n@\delta_\phi) \vdash \mathsf{agg}_\phi[\alpha](e_1,\ldots,e_m) : \mathsf{O}\,!\,\alpha}\ (\textsc{Agg})$$

Figure 6: The four rules that carry the accounting, extracted from the system below. CGEN's effect is the calibrated level, subsuming its generator, because the gate certifies its own output and a rejected draft contributes no deployment loss; AGG likewise. The LTT_Z antecedents are declared obligations, discharged at deployment rather than decided by type checking.

Yifan Xiao, Zhongjun Wang, Tiantian Yan, Weiwei Li, Dazhong Wu, and Jie Ma. Time series anomaly detection: A hybrid deep learning framework with adaptive thresholding for electricity industry data. In *IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)*, 2025. DOI 10.1109/CSCLOUD66326.2025.00046.

Yifan Xiao, Shijie Li, and Yuhao Ge. RunbookFX: Type- and effect-safe LLM synthesis for executable incident diagnosis and mitigation. *Proceedings of the ACM on Programming Languages*, 10(ICFP), 2026a. Article 277; DOI 10.1145/3828675.

Yifan Xiao, Shijie Li, and Yu Huang. CALM-TS: Risk-controlled LLM labeling for time-series via calibrated selective gating. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2026b. DOI 10.1145/3770855.3817859.

Dexuan Xu, Yanyuan Chen, Zhongyan Chai, Yifan Xiao, Yandong Yan, Weiping Ding, Hanpin Wang, Zhi Jin, Wenpin Jiao, Weihua Yue, Hang Li, and Yu Huang. Knowledge fusion in deep learning-based medical vision-language models: A review. *Information Fusion*, 125:103455, 2026. doi: 10.1016/j.inffus.2025.103455.

A Full Syntax and Typing Rules

$$\begin{aligned}
\tau &::= \mathsf{Unit} \mid \mathsf{O} \mid \tau_1 \to \tau_2 \mid \tau_1 \times \tau_2 \\
e &::= x \mid \langle\rangle \mid \lambda x.e \mid e\,e \mid \langle e, e\rangle \mid \pi_i\, e \\
&\quad \mid \mathsf{oracle}(e) \mid \mathsf{cgen}[\alpha](e_{\mathrm{gen}}, e_{\mathrm{fb}}) \mid \mathsf{agg}_\phi[\alpha](e_1,\ldots,e_m) \\
\Gamma &::= \cdot \mid \Gamma, x{:}\tau \qquad \Delta ::= \cdot \mid \Delta, (n\ \text{samples}@\delta)
\end{aligned}$$

O is the oracle/answer type and Δ is the linear calibration-budget context, a multiset of sample counts paired with confidence allowances. Beyond the rules of Figure 6, the variable rule types x at effect 0 with an empty budget, and APP, ABS and PAIR are the evident congruences: each sums the effects of its premises and merges their budgets with $\uplus$, the multiset union that denies contraction. Projections leave the effect unchanged. The absence of a contraction rule on Δ is the entire mechanism: a derivation cannot use the same calibration allowance in two premises, so the number of LTT_Z antecedents discharged in a derivation is bounded by the budget appearing in its conclusion.

A worked derivation. On the draft-then-check program $e = \mathsf{let}\ x = \mathsf{cgen}[\alpha_1](\mathsf{oracle}, \mathsf{abstain})\ \mathsf{in}\ \mathsf{cgen}[\alpha_2](x, \mathsf{abstain})$, the two gates type at their declared levels and LET sums the effects while merging the budgets, giving $\cdot;(n@\delta_1) \uplus (n@\delta_2) \vdash e : \mathsf{O}\,!\,\alpha_1{+}\alpha_2$, which is exactly what infer returns. The gate effects do not carry the oracle's effect 1 upward because CGEN subsumes its generator: the gate calibrates its own output risk, and a draft it rejects contributes no deployment loss. Two well-formedness checks are load-bearing. Were x consumed twice, single-consumer would fail and the static ledger would no longer match the runtime test count; were x routed back into the oracle, the feed-forward check would fail and Theorem 4 would apply.

B Loss Semantics and Proofs

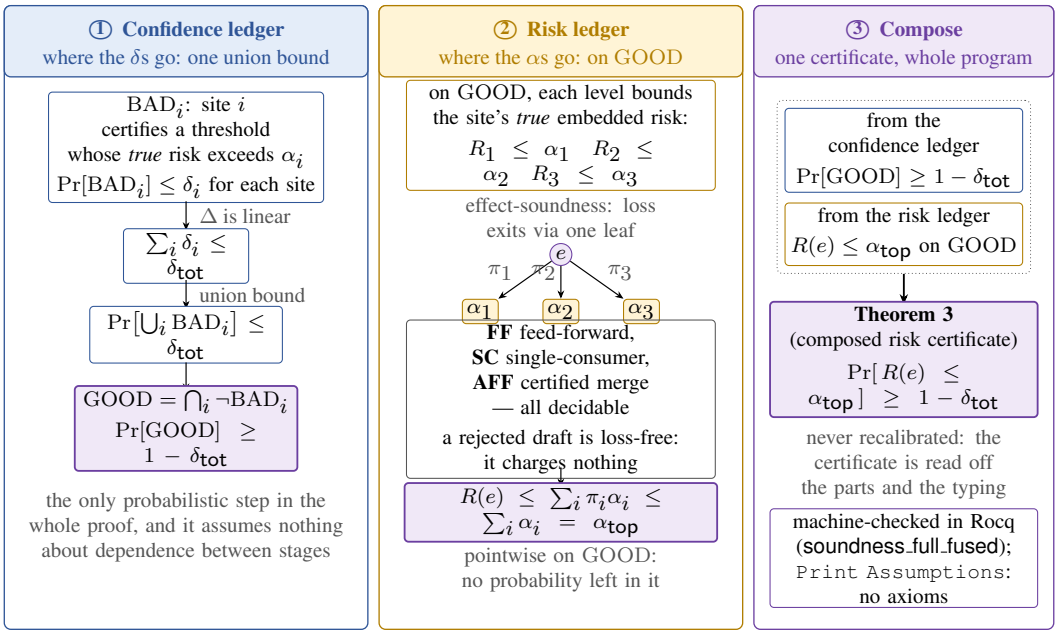


Figure 7: How Theorem 3 is assembled, in the order the proof below establishes it. The two ledgers do different jobs and meet only at the end. *Left:* each site's leaf certificate bounds one failure event BAD_i; linearity of Δ bounds how many such events there are, and a union bound, which needs no independence between stages, gives the good event GOOD probability at least $1 - \delta_{\text{tot}}$. This is the only probabilistic step in the development. *Middle: on* GOOD nothing is random any more. Each declared level is then a bound on the site's true embedded risk, and effect-soundness, which the three decidable well-formedness checks buy, forces every terminal loss path through exactly one calibrated leaf, so the output risk is the reach-weighted sum and is at most α_{top} pointwise. *Right:* composing a probability with a bound that holds on that event yields the certificate for the assembled program, with no recalibration of the whole. The fused capstone theorem of §5 checks this composition with $1 - \sum_i \delta_i$ in place of $1 - \delta_{\text{tot}}$.

Semantics. A closed program is interpreted against a measurable space of worlds $(\Omega, \mathcal{F}, P)$, where a world fixes the oracle draws and the unknown deployment input. Big-step evaluation $[\![e]\!]_\omega$ produces an output, the marker $L(\cdot) \in \{0, 1\}$ flags it, and $R(e) = \mathbb{E}_{\omega \sim P}[L([\![e]\!]_\omega)]$. An accepting gate emits its generator's value and a rejecting one emits the loss-free abstain, so $L(\text{cgen}[\alpha](e_g, e_f)) = acc_t([\![e_g]\!]_\omega) \wedge L([\![e_g]\!]_\omega)$; elimination forms propagate loss by disjunction; an aggregator's loss is that of its combined output, which its own certificate bounds directly. These clauses are the `Parameters` and derived lemmas of the Semantics module.

Proof of Lemma 1. Induction on the typing derivation. Every elimination rule sums the effects of its premises and every leaf rule contributes exactly its own effect, so the root effect is the leaf sum. Machine-checked as effect_soundness.

Proof of Lemma 2. Single-consumer forbids fan-out, so no leaf is charged twice; certified aggregation routes every merge through a node calibrating the composite; and the abstain convention makes a gate's reject branch loss-free. Hence loss at the output cannot be manufactured from individually-acceptable leaves and must come from one leaf emitting a bad value, whose risk is bounded by its level on the good event. Taking expectations and weighting by reach probability gives $R(e) \le \sum_i \pi_i \alpha_i \le \alpha_{\text{top}}$.

Proof of Theorem 3. Let BAD_i be the event that stage i certifies a threshold whose true risk exceeds its level; the cited leaf certificate gives $\Pr[\text{BAD}_i] \le \delta_i$. Linearity of Δ bounds the number

of stages, so $\sum_i \delta_i \le \delta_{\mathsf{tot}}$ and $\Pr[\bigcup_i \mathrm{BAD}_i] \le \delta_{\mathsf{tot}}$ by the union bound, with no independence assumption. On the complement, in-context calibration makes each stage's certified quantity its true embedded risk and Lemma 2 gives $R(e) \le \alpha_{\mathsf{top}}$ deterministically. Hence $\Pr[R(e) \le \alpha_{\mathsf{top}}] \ge 1 - \delta_{\mathsf{tot}}$. Rocq checks the probabilistic assembly and its composition with the deterministic bound, with no axioms, as soundness_full_fused, at $\mathbb{N}$-valued levels and with conclusion $1 - \sum_i \delta_i$. The step $\sum_i \delta_i \le \delta_{\mathsf{tot}}$ is the argument above and is not part of the Rocq statement.

C Mechanization Detail

Table 4 summarises the shipped development.

A note on identifiers. The Rocq sources name the abstract loss marker `harm`, from an earlier framing of this work; we keep the identifiers so the shipped development matches its captured transcripts, and the same name appears in the evaluation scripts for the same reason. The marker is an arbitrary bounded loss, and §6 instantiates it as answer error throughout, so nothing in the development or the tables bounds harm in any other sense.

Table 4: The Rocq development: `make`-green, 47 named results, no `Admitted` goals. union_bound, soundness_full, soundness_det and infer_sound close under the global context.

module	results	content
Core	9	syntax, typing, effect-soundness, budget, good-event bound (no axioms)
Semantics	8	loss semantics; derives composition and gating results
Gating	2	gate insertion as a sound transformation
Agg	5	certified aggregation (AGG-LIN subsumption)
Infer	4	decidable checking and principal inference (no axioms)
Prob	5	finite-probability model; union bound (no axioms)
Rat	7	effect/budget accounting and soundness over $\mathbb{Q}$, $[0, 1]$ (no axioms)
Wf	5	decidable well-formedness; tight ledger (no axioms)
Capstone	2	union bound with the deterministic core, conclusion $1 - \sum_i \delta_i$ (no axioms)

D Additional Evaluation Detail

Controlled pipelines. Before the real-model study we validate inference and calibration end to end on three feed-forward pipelines driven by a reproducible stochastic loss process: a single gate, a two-stage chain and a certified aggregation. At $\alpha = \delta = 0.05$ with $n_{\mathrm{cal}} = 2000$ and $n_{\mathrm{test}} = 5000$ these infer α_{top} of 0.05, 0.10 and 0.05 respectively, certify thresholds of 0.68, 0.72 and 0.70, retain accept rates of 0.326, 0.282 and 0.295, and realize out-of-sample risks of 0.0352, 0.0308 and 0.0292, all within their certified levels.

A note on "median". Every quantity the tables call a median is the upper of the two central order statistics (`sorted(xs)[len(xs)//2]`), not their average: the 101st of 200 split values and the 26th of 50 resampling runs. The distinction never moves a printed digit here, but the name is the looser of the two and the scripts use it throughout.

Table 2, re-generated. The generation runs behind Table 2 were performed once per cell without a sampling seed, so a certify/decline verdict there is in the first instance a statement about one draw of the generator rather than about the gate – most sharply for a cell whose unanimity error sits on the α=0.05 boundary. We re-generated all ten Qwen cells with `run_real_llm.py --out-suffix _rep2 --seed 20260828`, leaving the audited originals untouched beside them, and certified both members of each pair through the same calibrator at the same α, δ, threshold grid and split seed (`compare_rep2.py`; artifact `rep2_compare.json`).

Two arms are a pair of numbers rather than a spread. A rate that reads 51 on one generation and 200 on the next has moved, but nothing in two draws says how far it moves, and the first thing that sentence invites a reader to do – treat the difference as the size of the effect – is the thing two points cannot support. We therefore generated the ten Qwen cells

a *third* time, `--out-suffix _rep3 --seed 20260829`, and report all three together (`compare_reps.py`; artifact `reps_compare.json`). The read-out was fixed before those runs finished: the certifying frequency is reported as all three counts and their range and never as a mean or a dispersion estimate over three points; a verdict is a *pattern* over the arms, so only $3/3$ or $0/3$ is unanimous and anything else is named as split; and the quantity that must clear $1-\delta$ is the *marginal* coverage, in every arm, never the selection-conditional one. The three-way shared question set is the two-way one, $1{,}000$ ids per cell, so Table 10 and Table 7 may be read across.

What the third generation settles is the reading rather than the size. Four of the sixteen triples are not monotone – the certifying count falls and then rises, or rises and then falls – so no ordering of the runs makes the sequence a drift, and the two-point difference the second generation appeared to show was a difference between two draws. The widest spread is 149 of 200 on a cell whose verdict never moves, which is the same point from the other side: a cell can certify in every generation and still have its certifying *frequency* range over three quarters of the draws. Fourteen of the sixteen cells return one verdict in all three generations; the two exceptions: the cell whose unanimity error sits on α certifies in two of the three, so the audited original's decline is the minority of the three readings rather than a settled one, and Llama-8B/ARC certifies in one. Neither outcome is described as stable: at three generations only $3/3$ or $0/3$ would earn that word, and neither cell reaches either.

The budget's data-free prediction, checked on the added families. The bound that rules a cell out before any data is collected depends only on $(\alpha, n_{\mathrm{cal}}, \delta_{\mathrm{each}}, K)$, so it predicts the same outcome for every family. We took the 29 configurations it rules out for the Qwen cells and ran the certifying procedure under them on six pools – the four added-family cells and the two Qwen2.5-7B text cells – for 174 (configuration, pool) pairs. All 174 certify on 0 of 200 folds. The check is live: on three configurations the bound does not rule out, 15 of 18 pairs certify on at least one fold.

The grid's price, measured per cell (A-7). A-7 re-certifies all ten cells and all three generations on the grid $K{=}9 = \{0, \frac{1}{8}, \ldots, 1\}$, the vote shares $m{=}8$ chains can produce, with every other parameter of Table 2 unchanged; Table 5 gives both grids' counts for every cell and generation. The published table keeps $K{=}51$. The coarse grid certifies at least as often as the published one on all thirty cell-generations. Where it certifies strictly more is a per-cell answer, and it is reported per cell. On 7B/ARC and VL-3B/A-OKVQA nothing moves: 0 of 200 under both grids in every generation, and the draws carrying an admissible point estimate are the same under the two grids – 143, 111 and 148 on 7B/ARC, 198, 200 and 200 on VL-3B/A-OKVQA – so the correction refuses them at $\delta/9$ exactly as at $\delta/51$. The refusal of these two cells is not the grid's price. VL-7B/A-OKVQA is between: 0, 5 and 5 under $K{=}51$ become 1, 38 and 37 under $K{=}9$, strictly more in every generation and far short of the pre-registered rescue mark (half the splits). 3B/ARC is the most price-sensitive cell: 42, 30 and 115 become 149, 76 and 171, and its median threshold moves 0.88 to 1.0 in every generation. The original generations of the two ScienceQA cells move too, 54 to 146 and 141 to 198; both cells' replications already certify 200 of 200 under $K{=}51$. 1.5B/ARC does not move: 0 of 200 under both grids in every generation, and none of its declines carries an admissible point estimate for a coarser grid to free. The registered threshold prediction fails: the published medians 0.76 and 0.88 are not eighths, so $K{=}9$ moves the median threshold of every certifying cell to a reachable share – 0.875 on the GSM8K cells except the 1.5B replications, which take 0.75, and 1.0 on 3B/ARC, the ScienceQA cells and VL-7B/A-OKVQA. Where a cell already certifies 200 of 200, the count does not move; the threshold does. For 7B/ARC and VL-3B/A-OKVQA the grid's price is ruled out as the cause of refusal; the $m{=}8$ vote resolution is the candidate that remains, and B-3 tests it.

The vote-resolution sweep; a machine difference first (B-3). B-3 re-generates the three text boundary cells – 7B/GSM8K, 7B/ARC and 3B/ARC – on a second machine at $m \in \{4, 16, 32\}$, generation seed 20260913, plus an $m{=}8$ control on that machine (the on-box runs below), because the published $m{=}8$ cells come from the first machine and, in the original generation, an earlier engine release and no recorded seed. The registered rule reads the control first. On 3B/ARC the on-box control certifies 158 of 200, against 42 for the published cell and 30 and 115 for the two replications – recomputed on each file's own full split in file order, Table 2's convention, not Table 7's shared-thousand comparison – and the difference is outside the reported three-generation scatter (range 56). The other two cells show no machine difference: 0 against 0 on 7B/ARC and 200 against 200 on 7B/GSM8K. The sweep is therefore read inside the on-box series, and the first machine's

numbers above are a cross-machine reference. Table 6 gives every on-box count. The registered prediction was that $m \in \{16, 32\}$ rescues both ARC cells. It rescues one. On 3B/ARC the series is 0, 158, 200 and 200 at m=4, 8, 16, 32: full-consensus errors fall, 50, 28, 22 and 11, a threshold just below 1.0 becomes admissible, and the cell certifies in full at m=16 and m=32. On 7B/ARC the series is 0, 0, 0 and 0: full-consensus errors fall the same way, 68, 56, 49 and 45, and no fold certifies anyway. The controls behave as registered: m=4 certifies no more than the m=8 control on any cell, and 7B/GSM8K stays at 200 of 200 at every m (200, 200, 200 and 200), its full-consensus errors 24, 16, 11 and 11. Not both ARC cells are rescued – the negative reading the registration fixed, reported with the counts. Read with A-7, the causes separate. 3B/ARC moves under both: the grid's price alone lifts it to 149, 76 and 171 on the first machine's generations, and the finer resolution completes the rescue on-box. 7B/ARC moves under neither. VL-7B/A-OKVQA is partially price-sensitive under A-7; the sweep covers the text cells only, so both directions stand reported for it.

Generations four and five on a second machine (pre-registered B-4). The ten Qwen cells were generated twice more under seeds 20260907 and 20260908 on an RTX 3090 (CUDA 12.1; the first three generations ran on an A800, CUDA 12.4), same protocol and pins; the four vision-language cells ran with vLLM's batch cap lowered to 16 from its default 256 (24 GB cards), and vLLM sampling is not batch-invariant, so those cells differ from their A800 counterparts in batch cap as well as in seed and machine. Table 9 lists the five counts per cell. Against the majority verdict of generations one to three, generation four agrees on 9 of 10 cells and generation five on 8; the certifying count lies inside the first three generations' range in 9 and 8 cells. 8 cells keep one verdict throughout; VL-7B/A-OKVQA certifies in $2/5$ (generations two and three only) and is reported as unstable, VL-3B/A-OKVQA in $1/5$. The widest five-generation range is 149 of 200 (VL-3B/ScienceQA); marginal guarantee coverage stays at 0.985 or above in all arms. The three-generation, one-machine spread above remains the paper's seed-variance figure; this five-generation figure carries the machine term.

The dependent chains of RQ2 were re-certified on every generation as well; Table 8 lists both chains' counts per generation, so the published 3B→7B count and the five-generation range the body quotes are one column read at one row and over all five.

The pair cannot be compared file against file. The original runs stopped at the first 1,000 questions of each test split; the replication loaded each split in full – 1,172 for ARC, 1,319 for GSM8K, 1,145 for A-OKVQA and 2,017 for ScienceQA. Certifying the two files as they stand would confound the quantity under test – what a second run changes – with a larger sample, so each pair is certified on the 1,000 question ids the two runs share, in question-id order, and the replication's own full-sample certificate is certified in that same order and reported beside the comparison rather than as it. Table 10 gives all three certificates of all ten Qwen cells for the first two generations.

What separates the two arms is not the sampling seed alone, and the difference should not be attributed to sampling randomness. The six text cells were generated in June under `vllm==0.6.3.post1`; the replication ran under `vllm==0.7.3` with `torch==2.5.1+cu121` and `transformers==4.57.6`, the stack the machine moved to before the vision–language cells were produced. The four vision–language cells are on `0.7.3` in both arms; the six text cells differ in engine release – sampler, kernels and chat-template handling – as well as in seed. The finding is not weakened by this: a verdict that does not survive a change of inference stack is no more a property of the gate than one that does not survive a change of seed. Both pins, and the launcher that produced the replication, ship with the artifact.

Between the first two generations, nine of the ten verdicts reproduce and the exception sits on the boundary; the third generation's reading is above. Both ends of Table 2 reproduce exactly: the three GSM8K cells certify on all 200 draws in both runs, and 1.5B/ARC, 7B/ARC and VL-3B/A-OKVQA certify on none in either. One verdict of the ten flips: VL-7B/A-OKVQA goes from no certifying split in the original to 7 of 200 in the replication. 3B/ARC certifies in both but on 10 of 200 splits against 41, and one of those ten draws sees its out-of-sample risk exceed α where none of the 41 did. 7B/ARC declines in both, and does so although its base error *fell*, 0.1100 to 0.1010 on the shared questions: a decline is not a function of the base rate alone. What moved underneath is small. Base error moves by half a point or less on identical questions, 0.1400 to 0.1440 at 3B/ARC and 0.1270 to 0.1230 at VL-7B/A-OKVQA, from 42 and 38 of the shared questions changing loss label. We

report the four ‡ rows of Table 2 as not distinguishable at this sample size and rest no claim on which side of the boundary they fell.

The certifying *rate* is the unstable quantity, and it is least stable on cells whose verdict never moves. Over the three generations VL-3B/ScienceQA certifies on 51, 200 and 200 of 200 draws, VL-7B/ScienceQA on 143, 176 and 143, and 3B/ARC on 41, 10 and 66 – ranges of 149, 33 and 56. Two of those three sequences fall and then rise or rise and then fall, so no ordering of the runs makes any of this a trend, and the two-point differences the second generation appeared to show – a rate that had "become" another rate – were differences between draws. We therefore report all three counts and their range and take no mean over them: three points locate a spread, they do not estimate one. A $k/200$ printed for one of these cells is a property of one generation run, not of the gate, and the † counts of Table 2 are to be read that way. The verdict is the coarser statistic: it is unanimous across all three generations everywhere except where the gate sits on α.

The two coverage columns must not be read against each other, and only one of them may be read against $1 - \delta$. Where the replication certifies on a small minority of folds – 10, 23, 7 and 1 of 200, at 3B/ARC and VL-7B/A-OKVQA – "emp. guar." is a frequency over ten draws or fewer, and it is computed *given* that a certificate was issued. The guarantee is stated marginally over the calibration draw: a draw on which no threshold is returned claims nothing and so violates nothing, and the quantity that must sit at or above $1 - \delta = 0.95$ is the marginal column. It does: 0.990 at worst across the thirty arms here, 0.980 at worst across all ninety-six arms of the three generations, and 0.975 at worst across the certifying cells of Table 2, against a target of 0.95. *No arm of Table 10 misses its guarantee.* What the replication moves is how often a certificate is issued at all, and that, not a coverage shortfall, is the finding. The same distinction governs the $\alpha = 0.02$ cells of the margin-sweep replication below, where three certifying draws support a conditional rate and $3{,}000$ support the marginal one.

One artifact of fixing the question order is worth naming so Table 10 is not read as contradicting Table 2. All arms of Table 10 certify in question-id order, while Table 2 certifies each cached file in the order it was written; `certify_one` shuffles a list, so under one split seed those are two different assignments of the same questions. The two orders agree exactly wherever a cell certifies on all 200 draws or on none; across the three cells of Table 2 that certify on some but not all, the two orders differ by one to three draws – 3B/ARC 41 against 42, VL-3B/ScienceQA 51 against 54, VL-7B/ScienceQA 143 against 141 – and by as much as seven in the full-sample replication arms. That is the ordering, not the data, and it is a further reminder of how little separates these cells from declining.

The full margin sweep. RQ4 summarizes a sweep over $\alpha \in \{0.02, 0.03, 0.05, 0.08\}$, calibration-fold size $n_{\mathrm{cal}} \in \{100, 200, 400, 700\}$ and stage count $k \in \{2, 3, 6\}$, 48 cells in total, each over 300 splits, comparing the family-wise violation rate with the budget correction against without. What the sweep calls a violation is the event $\hat{R}_{\mathrm{test}} > \alpha$ on the held-out fold, not $R > \alpha$: an upper-biased proxy for the quantity δ bounds, and one whose granularity is $1/n_{\mathrm{test}}$ against a target of α. The pools hold $1{,}000$ questions, so the test folds are 900, 800, 600 and 300 items across the n_{cal} grid, and at $n_{\mathrm{cal}} = 700$ the indicator resolves α only to $1/300$. Near the boundary, the thin-margin regime this sweep probes, it therefore fires close to half the time whether or not the certificate holds. Both conditions are affected alike, so the comparison between them survives; the levels should not be read as estimates of the true violation probability. The k gates are the GSM8K leaves in increasing model size, extended by the ARC leaves at $k = 6$; the sweep is over the text leaves only and the vision–language cells of Table 2 do not enter it, so the counts below are unaffected by them (the replication at the end of this section lifts that restriction).

Each condition draws all k calibration folds for a split *before* calibrating any gate, so the corrected and uncorrected runs of a cell consume one random stream at the same rate and see the same sequence of splits however early either of them declines. This matters: an earlier version drew each fold inside the calibration loop and broke out of it on the first gate that declined, which skipped the remaining draws, desynchronised the two streams from the first split on which the conditions disagreed, and turned every "corrected against uncorrected" count in this section into a comparison of two different Monte-Carlo samples. All counts below are from the paired run.

Which cells could have certified at all. Thirty-one cells certify on no split at all, and *fifteen of those thirty-one could not have certified on any data*. Certification requires a grid threshold whose Hoeffding–Bentkus p-value is at most δ_i/K, and the smallest p-value any fold can attain is the zero-empirical-risk value $(1-\alpha)^{n_{\text{cal}}}$, so a cell certifies only if $n_{\text{cal}} \geq \ln(\delta_i/K)/\ln(1-\alpha)$ – at $k = 1$ that is 343, 227, 135 and 83 for $\alpha = 0.02$, 0.03, 0.05 and 0.08, each about a third larger at $k = 10$. Against the grid $n_{\text{cal}} \in \{100, 200, 400, 700\}$ that rules out every $\alpha \in \{0.02, 0.03\}$ cell at $n_{\text{cal}} \in \{100, 200\}$ and the three $\alpha = 0.05$, $n_{\text{cal}} = 100$ cells: 15 of the 48 here, 20 of the 64 in the replication, all of them among the cells that certify on no split. Two further cells ($\alpha = 0.02$, $n_{\text{cal}} = 400$, $k = 6$ and $\alpha = 0.08$, $n_{\text{cal}} = 100$, $k = 6$; four in the replication) are impossible for the corrected condition alone, so there the correction costs certification power by arithmetic before any data is seen. Figure 8 draws the deadline over the sweep.

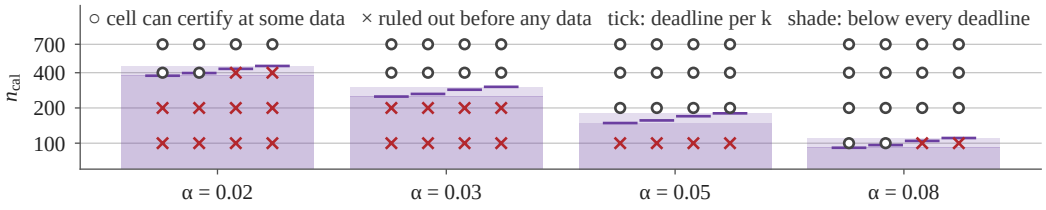


Figure 8: The certification deadline reads off the configuration. For each α of the 64-cell sweep, the tick at each k is the smallest n_{cal} at which a zero-empirical-risk fold can clear δ_i/K, i.e. $\ln(\delta_i/K)/\ln(1-\alpha)$, as `margin_sweep.py` records it before any run file is opened; the shade is the region below every deadline. Markers are the sweep's cells at $n_{\text{cal}} \in \{100, 200, 400, 700\}$: × marks the 20 ruled out at any data, which by this arithmetic certify on no split.

These are not empirical outcomes and we do not count them as such: the feasible denominators are 33 of 48 and 44 of 64, and the counts below are quoted against the seventeen and twenty cells that certify. `margin_sweep.py` records the bound and a per-cell `feasible_at_all` flag so this arithmetic is checkable.

The $k = 6$ configurations never certify for a different, empirical reason: they include the ARC leaves that RQ1 shows declining. Of the seventeen cells that do certify, the two conditions differ on eight in one dimension or the other and on two in both: four post a different family-wise error in the two conditions, and six certify on different numbers of splits. `margin_sweep.py` records a per-cell boolean under a narrower criterion – uncorrected family-wise error strictly above δ while corrected is at or below it – and under *that* criterion exactly one cell separates them, at $\alpha = 0.03$, $n_{\text{cal}} = 700$, $k = 3$: the uncorrected condition certifies on 83 splits and violates on 5 of them (0.060), the corrected condition certifies on 19 and violates on none (0.000). A two-sided 95% Clopper–Pearson interval on $5/83$ – exact binomial, 0.025 of tail probability on each side – is $[0.020, 0.135]$ and contains the 0.050 target: not resolvable at this split count. The other three differing cells, all at $n_{\text{cal}} = 700$, keep both conditions under δ: at $\alpha = 0.03$, $k = 2$ the uncorrected condition violates on 5 of 117 certified splits (0.043) against 2 of 58 (0.034); at $\alpha = 0.08$, $k = 2$, on 5 of 300 (0.017) against 1 of 300 (0.003); at $\alpha = 0.08$, $k = 3$, on 8 of 300 (0.027) against 3 of 300 (0.010). The corrected condition posts the lower rate in all four, the direction the theory predicts, but no one of the four differences is resolvable at 300 splits (smallest two-sided Fisher $p = 0.22$), and no cell in the sweep puts the uncorrected family-wise error significantly above δ – the largest, $5/83$, gives a one-sided exact $p = 0.40$ against $\delta = 0.05$. What the correction costs is certification power: six of the seventeen cells certify on different numbers of splits, by 19 against 83 at $\alpha = 0.03$, $k = 3$ and by 187 against 253 at $\alpha = 0.05$, $n_{\text{cal}} = 400$, $k = 3$. Four of those six move only the split count – $\alpha = 0.05$ with $n_{\text{cal}} = 400$ and $\alpha = 0.08$ with $n_{\text{cal}} = 200$, each at $k = 2$ and $k = 3$ – and in all four both conditions post *zero* violations, so the correction posts strictly fewer only where the family-wise error differs. The sweep records violation rates and split counts only, never the selected thresholds, so it supports no claim about whether the two conditions pick the same threshold; for the stage sweep of Table 13, where we did replay the picks, they do not always. The raw per-cell records ship with the artifact.

The sweep replicated at ten times the splits. Two counts above are left open by 300 splits: whether the single separating cell reflects a real excess, and whether the four rate differences have a

direction that replicates. Re-running the sweep at 3,000 splits per cell, with the four vision–language leaves of Table 2 appended to the pool set so that k=10 joins the stage grid (64 cells; artifact `margin_sweep_v2.json`), answers both. The separating cell does not replicate: at $\alpha = 0.03$, $n_{\text{cal}} = 700$, $k = 3$ the uncorrected condition certifies on 809 of 3,000 splits and violates on 28 of them (0.035), below the 0.050 target, against 5 of 229 (0.022) corrected, and no cell among the 20 that certify satisfies the biting criterion. The direction, by contrast, resolves: six of the 20 cells that certify carry a different family-wise error in the two conditions – the four from the original sweep joined by the two $\alpha = 0.05$, $n_{\text{cal}} = 400$ cells – the corrected condition lower in all six, with two-sided Fisher p of 0.0285 (7/636 against 34/1272), 0.40 (5/229 against 28/809), 0.50 (0/2237 against 2/2669), 1.0 (0/1896 against 1/2543), 0.029 (9/3000 against 22/3000) and 0.0014 (25/3000 against 54/3000); the last, at $\alpha = 0.08$, $k = 3$, survives a Holm correction over the six ($0.0013643 \times 6 = 0.0082$, from the unrounded p) and no other does. These are per-cell p-values on one Monte-Carlo run and should be read as such: the split counts they condition on move by a few percent under a different split seed, and the 0.40 moves furthest with them. The direction – corrected lower in all six – and the Holm-surviving cell, which certifies on all 3,000 splits in both conditions, are the stable parts. The certification-power cost survives replication and sharpens – 229 against 809 splits at $\alpha = 0.03$, $k = 3$ – and the two cells that differ only in split count, both at $\alpha = 0.08$, $n_{\text{cal}} = 200$, post zero violations in both conditions (2,491 against 2,864 and 2,188 against 2,762 splits certified). Three features of the replication need stating so its counts cannot mislead. First, three cells certify at 3,000 splits that never did at 300: one at $\alpha = 0.03$, $n_{\text{cal}} = 400$, $k = 3$, and the two $\alpha = 0.02$, $n_{\text{cal}} \in \{400, 700\}$, $k = 2$ cells, which certify on 10 and 3 splits and post conditional violation rates of 1/10 and 3/3 – identically in the two conditions. Those rates are an artifact of conditioning on rare certification, and the sweep now measures the conditioning rather than asserting it: at $\alpha = 0.02$, $n_{\text{cal}} = 700$ the mean least calibration risk any grid threshold could post is 0.0029 on the three folds that certify against 0.0111 on the 2,997 that do not, a factor of nearly four. Only the luckiest folds certify, and their thresholds are the ones that lie. The guarantee the certificate states is marginal over the calibration draw – violate with probability at most δ – and the marginal rates, 1 and 3 violations in 3,000 splits, sit far inside either condition's budget; a conditional rate on 3 splits separates nothing, and it does not differ between the conditions it would need to separate.

Second, that cell is the one place in this paper where a conditional violation rate is significantly above δ, and we state it here rather than only where it is convenient. Three of three certifying draws violate at $\alpha = 0.02$, $n_{\text{cal}} = 700$, $k = 2$; a one-sided exact binomial test of 3/3 against $\delta = 0.05$ gives $p = 1.25 \times 10^{-4}$. So the 48-cell sweep's headline – no cell puts the uncorrected family-wise error significantly above its target – does *not* extend verbatim to the 64-cell replication, and §6 says so. What defends the certificate is the marginal reading above, not the absence of the cell: the same 3/3 appears in *both* conditions, so it separates nothing, and 3 violations in 3,000 draws is a marginal rate of 0.001.

Third, no k=10 configuration certifies on any split, for the reason the k=6 text cells already showed: the stage set contains the declining leaves, and a chain with an uncertifiable stage is uncertifiable, so the vision–language extension widens the grid without adding certifying cells. The replication script is the shipped `margin_sweep.py` with `--nsplits 3000 --leaves all`; the original 48-cell artifact is unchanged beside it.

The body figures, in full. The three tables and one figure below carry the exact entries the body plots. Nothing here is a further result: Table 11 is the RQ3 comparison whose first four columns are Figure 1, plus the per-leaf calibration cost the typed row has already paid; Table 13 is the grid and stage sweep of Figure 5a at printed precision, with the median coverage the figure drops; and Figure 9 is the expanded version of Figure 2, drawing the per-site Learn-then-Test step in full.

E THE CLOSED-LOOP IMPOSSIBILITY ARGUMENT

This section proves Theorem 4, whose hypotheses *characterize* the loop dynamics the two-point (Le Cam) argument needs rather than exhibiting a pair: what follows is a proof of the implication, not an existence proof. The theorem is proved on paper rather than mechanized: closed loops are not expressible in the acyclic-by-construction syntax the development checks (§5).

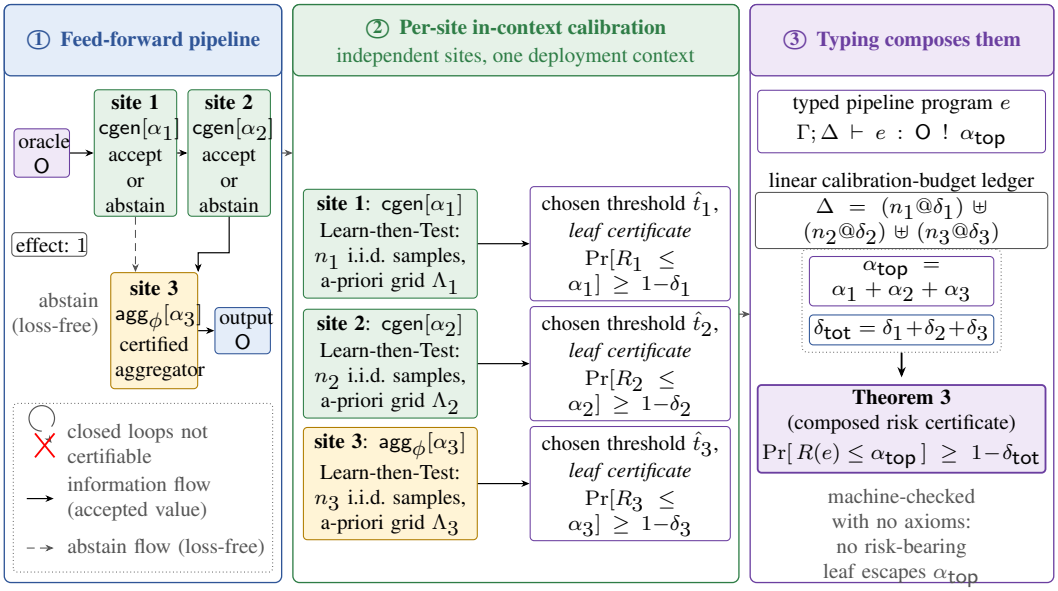


Figure 9: λ-Cert end to end, with the calibration step Figure 2 compresses. *Left:* a feed-forward pipeline of gates and a certified aggregator, closed loops marked as outside the certifiable fragment (Theorem 4). *Middle:* each site is calibrated in-context by Learn-then-Test against an a-priori grid, discharging its leaf certificate. *Right:* the typing judgment threads the level and the linear calibration budget through the program, and Theorem 3 assembles the per-site certificates into $\Pr[R(e) \leq \alpha_{\mathsf{top}}] \geq 1 - \delta_{\mathsf{tot}}$.

Setting. Fix $\alpha \in (0,1)$, a single gate with accept threshold $t \in [0,1]$, and a utility floor $\rho > 0$, the gate having to accept at least a ρ fraction; this is the nonzero acceptance rate the theorem requires, and it excludes always abstaining. A *static calibrator* draws a calibration sample $S \in \mathcal{Z}^n$ from the pre-deployment law $\Phi(t_0)$, t_0 being the pre-deployment policy, selects a threshold through a measurable map $\hat{t} : \mathcal{Z}^n \to [0,1]$ ($\mathcal{Z}^n$ carrying its product σ-algebra and $[0,1]$ the Borel σ-algebra) – or, if randomized, through such a map of S and an independent seed U – and deploys the selected value, under which the input law becomes $P = \Phi(\hat{t})$ as in §4.

Proof of Theorem 4. Let Φ_0, Φ_1 be dynamics satisfying the theorem's two hypotheses. The first is indistinguishability: they are *identical* at t_0, so S has the *same* law μ under both. The second is adversarial placement: for every threshold value t whose accepted region has mass at least ρ, the loss mass under at least one Φ_b lies on the accepted side, so that $R_{\Phi_b(t)}(t) > \alpha$ for that b. Write $e_b = \Pr_{S\sim\mu}[R_{P^b}(\hat{t}) > \alpha]$ for the per-world error rate of a fixed selector. Adversarial placement is a property of the deployed threshold *value*, not of how it was chosen, and every value $\hat{t}$ takes accepts mass at least ρ by the theorem's floor, so $\mathbf{1}[R_{P^0}(\hat{t}) > \alpha] + \mathbf{1}[R_{P^1}(\hat{t}) > \alpha] \geq 1$ holds pointwise in the realization; and S has one law across the two worlds, so taking expectations gives $e_0 + e_1 \geq 1$ and hence $\max_b e_b \geq \frac{1}{2}$, that is $\sup_b \Pr[R_{P^b}(\hat{t}) > \alpha] \geq \frac{1}{2}$. Nothing in the step assumes $\hat{t}$ is a deterministic function of S: a randomized selector is a measurable function of (S,U), the pointwise inequality holds at each (S,U), and the same expectation gives the bound. It holds at every sample size n, since n enters only through μ.

A pair that satisfies both hypotheses. The implication is not vacuous, and what defeats the calibrator is the indistinguishability rather than a conflict between the floor and the target inside one world.

Proposition 5 (A certifiable, indistinguishable pair) *Fix $\alpha \in (0,1)$ and $\rho \in (0,1)$ with $\alpha < \min\{\rho, (1-\rho)/2\}$. There exist closed-loop dynamics Φ_0, Φ_1 that satisfy both hypotheses of Theorem 4 and such that, for each b, some threshold accepts mass at least ρ with $R_{\Phi_b(t)}(t) \leq \alpha$.*

Proof. Let $\mathcal{Z} = [0,1] \times \{0,1\}$ carry a score x and a loss marker L, the gate accepting iff $x \geq t$, so a threshold t accepts mass $1 - t$ and the floor allows $t \leq 1 - \rho$. Pick m with $\alpha < m <$

$\min\{\rho, (1-\rho)/2\}$ and set $t^\star = (1-\rho)/2$. Under every law below, x is uniform on $[0,1]$ and $L = 1$ exactly on one interval of length m; abstention is loss-free, so $R_{\Phi_b(t)}(t)$ is the mass of that interval inside $[t,1]$. Both worlds share the pre-deployment law $\Phi_b(t_0)$, $t_0 = 1$, with the interval $[0,m]$. Under $\Phi_0(t)$ the interval is $[t, t+m]$ for $t < t^\star$ and $[0,m]$ for $t \geq t^\star$; under $\Phi_1(t)$ it is $[t, t+m]$ for $t < m$ and for $t^\star \leq t \leq 1-\rho$, and $[0,m]$ for every other t, including t_0. The two laws coincide at t_0, so a calibration sample has one law under both. For every feasible $t \leq 1-\rho$ the interval $[t, t+m]$ lies inside the accepted region ($t + m \leq 1 - \rho + m \leq 1$) and carries risk $m > \alpha$: in world 0 when $t < t^\star$, in world 1 when $t \geq t^\star$, so placement is adversarial. Whenever the interval is $[0,m]$ it lies inside the rejected region ($m \leq t$) and the risk is 0: world 0 is certified by any $t \in [t^\star, 1-\rho]$, which accepts mass at least ρ, and world 1 by any $t \in [m, t^\star)$, which accepts mass above $1 - t^\star \geq \rho$. Theorem 4 then gives $\sup_b \Pr[R_{P^b}(\hat{t}) > \alpha] \geq \frac{1}{2}$ for every static calibrator that meets the floor, while an oracle that knows b certifies either world. □ Mechanizing the argument is open. The escape the implication points at – iterating to a fixpoint when Φ is contractive – breaks the indistinguishability by calibrating on-policy at convergence.

Table 5: The grid's price, per cell and generation (A-7). Splits of 200 issuing a certificate, the three generations in order, on the published $K{=}51$ grid and on $K{=}9 = \{0, 1/8, \ldots, 1\}$, the vote shares $m{=}8$ chains produce; every other parameter of Table 2 is unchanged. $\hat{t}$ is the median certified threshold, printed once where the generations agree; "–": no split certifies. $K{=}9$ certifies at least as often as $K{=}51$ on all thirty cell-generations. The published table keeps $K{=}51$.

cell	certified, $K{=}51$	certified, $K{=}9$	$\hat{t}$, $K{=}51$	$\hat{t}$, $K{=}9$
1.5B/GSM8K	200, 200, 200	200, 200, 200	0.76	0.875, 0.75, 0.75
3B/GSM8K	200, 200, 200	200, 200, 200	0.76	0.875
7B/GSM8K	200, 200, 200	200, 200, 200	0.76	0.875
1.5B/ARC	0, 0, 0	0, 0, 0	–	–
3B/ARC	42, 30, 115	149, 76, 171	0.88	1.0
7B/ARC	0, 0, 0	0, 0, 0	–	–
VL-3B/ScienceQA	54, 200, 200	146, 200, 200	0.88	1.0
VL-7B/ScienceQA	141, 200, 200	198, 200, 200	0.88	1.0
VL-3B/A-OKVQA	0, 0, 0	0, 0, 0	–	–
VL-7B/A-OKVQA	0, 5, 5	1, 38, 37	–, 0.88, 0.88	1.0

Table 6: The vote-resolution sweep (B-3), on-box series. The three text boundary cells re-generated on the second machine at $m \in \{4, 8, 16, 32\}$ chains per question – $m{=}8$ is the on-box control, the published $m{=}8$ cells being the first machine's – under generation seed 20260913, certified as in Table 2. *errors*: wrong deployed answers of the cell's 1,172 (ARC) or 1,319 (GSM8K) questions; *full-consensus errors*: the wrong answers every chain agrees on, the bucket a finer m can split. The first machine's cells are not rows of this table; the registered rule reads the machine comparison first, and the text carries it.

cell	m	certified (of 200)	errors	full-consensus errors
7B/GSM8K	4	200	93	24
	8	200	82	16
	16	200	84	11
	32	200	87	11
7B/ARC	4	0	124	68
	8	0	120	56
	16	0	114	49
	32	0	110	45
3B/ARC	4	0	189	50
	8	158	179	28
	16	200	181	22
	32	200	173	11

Table 7: The sixteen cells of Table 2 across *three* independent generations (the audited original, `__rep2` and `__rep3`), each certified in question-id order through one calibrator at $\alpha{=}\delta{=}0.05$, 200 splits and the split seed of Table 2, on the 1,000 question ids all three share. *certifying draws* lists all three counts and their range: the frequency is a property of a draw, not of the cell, and three points are reported as three points – no mean and no dispersion estimate is taken over them. *verdict* is the pattern over the arms; $*$ marks the two cells that are not unanimous. *marg. cov.* is the smallest marginal coverage among that cell's six certificates (three generations $\times$ paired and own-sample); it, not the selection-conditional rate, is what $1-\delta$ bounds.

cell	certifying draws (of 200)	range	verdict	marg. cov.
1.5B/ARC	0, 0, 0	0	0/3	1.000
Llama-8B/ARC	4, 0, 0	4	1/3 $*$	0.980
Llama-8B/GSM8K	200, 200, 200	0	3/3	0.995
OLMo-7B/ARC	0, 0, 0	0	0/3	1.000
OLMo-7B/GSM8K	200, 200, 199	1	3/3	0.995
1.5B/GSM8K	200, 200, 200	0	3/3	0.995
3B/ARC	41, 10, 66	56	3/3	0.995
3B/GSM8K	200, 200, 200	0	3/3	1.000
7B/ARC	0, 0, 0	0	0/3	1.000
7B/GSM8K	200, 200, 200	0	3/3	0.990
14B/ARC	1, 14, 6	13	3/3	0.985
14B/GSM8K	200, 200, 200	0	3/3	0.985
VL-3B/A-OKVQA	0, 0, 0	0	0/3	1.000
VL-3B/ScienceQA	51, 200, 200	149	3/3	1.000
VL-7B/A-OKVQA	0, 7, 3	7	2/3 $*$	0.990
VL-7B/ScienceQA	143, 176, 143	33	3/3	0.995

Table 8: RQ2's dependent chains across the five generations, one row each. Both chains run at $\alpha{=}\delta{=}0.05$ per leaf (so $\alpha_{\mathsf{top}}{=}\delta_{\mathsf{tot}}{=}0.10$) over the same 200 splits and the split seed 20260622; the generations differ in the cached generation records the leaves are certified on, not in the protocol. Generation 1 is the published one: its 3B $\to$ 7B count is the row Table 3 and Table 11 print and Figure 1b plots, and the range down that column is what the five-generation audit reports. *median realized* and *holds* are the 3B $\to$ 7B chain's, on the splits where it certifies.

gen.	box	3B $\to$ 7B cert.	median realized	holds $\leq \alpha_{\mathsf{top}}$	1.5B $\to$ 3B cert.
1	A800	41/200	0.0152	100%	0/200
2	A800	31/200	0.0222	100%	5/200
3	A800	25/200	0.0230	100%	1/200
4	RTX 3090	42/200	0.0115	100%	33/200
5	RTX 3090	54/200	0.0114	100%	43/200

Table 9: The ten Qwen cells of Table 2 across *five* independent generations (the audited original, `__rep2`, `__rep3`, `__rep4` and `__rep5`), each certified in question-id order through one calibrator at $\alpha{=}\delta{=}0.05$, 200 splits and the split seed of Table 2, on the 1,000 question ids all five share. Generations 1–3 ran on the A800 (cu124); generations 4–5 ran on an RTX 3090 (cu121); the grouped header names the box each column ran on, so a box term is never read as seed variance. *verdict* is the pattern over the five arms; $*$ marks the two cells that are not unanimous. *marg. cov.* is the smallest marginal coverage among that cell's ten certificates (five generations $\times$ paired and own-sample); it, not the selection-conditional rate, is what $1-\delta$ bounds.

	A800			RTX 3090				
cell	1	2	3	4	5	range	verdict	marg. cov.
1.5B/ARC	0	0	0	0	0	0	0/5	1.000
1.5B/GSM8K	200	200	200	200	200	0	5/5	0.995
3B/ARC	41	10	66	8	11	58	5/5	0.995
3B/GSM8K	200	200	200	200	200	0	5/5	0.985
7B/ARC	0	0	0	0	0	0	0/5	1.000
7B/GSM8K	200	200	200	200	200	0	5/5	0.985
VL-3B/A-OKVQA	0	0	0	0	1	1	1/5 $*$	0.995
VL-3B/ScienceQA	51	200	200	140	77	149	5/5	1.000
VL-7B/A-OKVQA	0	7	3	0	0	7	2/5 $*$	0.990
VL-7B/ScienceQA	143	176	143	156	190	47	5/5	0.995

Table 10: All ten Qwen cells of Table 2 under the second, seeded generation run; the third generation is in Table 7. All three arms of every cell are certified *in question-id order* through the calibrator, grid, $\alpha{=}\delta{=}0.05$, 200 splits and split seed of Table 2, so "original" and "replication (shared)" – both on the 1,000 question ids the two runs share – differ only in which generation produced the scores, and "replication (full)", on the whole split the replication loaded, differs from it only in sample. "splits cert." counts the calibration draws on which some grid threshold certifies. The last two coverage columns are *different quantities*: "emp. guar." is the fraction of the *certifying* draws whose out-of-sample risk respected α, a selection-conditional frequency, undefined where none certify; "marg." is the fraction of all 200 draws that did not violate α, counting a draw that returns no threshold as satisfying the guarantee vacuously. It is "marg." that $1-\delta$ bounds.

cell	arm	n	certified	splits cert.	emp. guar.	marg.	base err.
1.5B/ARC	original	1,000	no	0/200	–	1.000	0.2660
1.5B/ARC	replication (shared)	1,000	no	0/200	–	1.000	0.2620
1.5B/ARC	replication (full)	1,172	no	0/200	–	1.000	0.2790
1.5B/GSM8K	original	1,000	yes	200/200	0.995	0.995	0.2230
1.5B/GSM8K	replication (shared)	1,000	yes	200/200	1.000	1.000	0.2170
1.5B/GSM8K	replication (full)	1,319	yes	200/200	0.995	0.995	0.2252
3B/ARC	original	1,000	yes	41/200	1.000	1.000	0.1400
3B/ARC	replication (shared)	1,000	yes	10/200	0.900	0.995	0.1440
3B/ARC	replication (full)	1,172	yes	23/200	0.957	0.995	0.1476
3B/GSM8K	original	1,000	yes	200/200	1.000	1.000	0.1120
3B/GSM8K	replication (shared)	1,000	yes	200/200	1.000	1.000	0.1130
3B/GSM8K	replication (full)	1,319	yes	200/200	0.995	0.995	0.1084
7B/ARC	original	1,000	no	0/200	–	1.000	0.1100
7B/ARC	replication (shared)	1,000	no	0/200	–	1.000	0.1010
7B/ARC	replication (full)	1,172	no	0/200	–	1.000	0.0964
7B/GSM8K	original	1,000	yes	200/200	0.990	0.990	0.0700
7B/GSM8K	replication (shared)	1,000	yes	200/200	1.000	1.000	0.0630
7B/GSM8K	replication (full)	1,319	yes	200/200	0.995	0.995	0.0622
VL-3B/A-OKVQA	original	1,000	no	0/200	–	1.000	0.1620
VL-3B/A-OKVQA	replication (shared)	1,000	no	0/200	–	1.000	0.1650
VL-3B/A-OKVQA	replication (full)	1,145	no	0/200	–	1.000	0.1668
VL-3B/ScienceQA	original	1,000	yes	51/200	1.000	1.000	0.2020
VL-3B/ScienceQA	replication (shared)	1,000	yes	200/200	1.000	1.000	0.2080
VL-3B/ScienceQA	replication (full)	2,017	yes	200/200	1.000	1.000	0.2038
VL-7B/A-OKVQA	original	1,000	no	0/200	–	1.000	0.1270
VL-7B/A-OKVQA	replication (shared)	1,000	yes	7/200	0.857	0.995	0.1230
VL-7B/A-OKVQA	replication (full)	1,145	yes	1/200	1.000	1.000	0.1249
VL-7B/ScienceQA	original	1,000	yes	143/200	1.000	1.000	0.0950
VL-7B/ScienceQA	replication (shared)	1,000	yes	176/200	0.994	0.995	0.1000
VL-7B/ScienceQA	replication (full)	2,017	yes	198/200	0.995	0.995	0.0972

Table 11: RQ3, in full; Figure 1 plots the first four columns. Certifying the 3B→7B pipeline by typing against black-box resampling with an exact Clopper–Pearson upper bound at confidence $1-\delta$, $\delta = 0.05$. "Draws" counts end-to-end runs scored *for this pipeline*, and again for any other; here drawn with replacement from the cached $1{,}000$-question pool (50 resampling experiments per row, median bound), so the resampled rows are measurements whose coverage Table 12 reports. "Returns" is how often a bound is returned at all. "Leaf labels" is what the typed row has *already* spent, once, per calibrated leaf: it is not zero, and it is not re-spent by the next pipeline built over the same leaves. The resampled pipeline fixes both thresholds at 0.76 under a two-stage OR loss, so its numbers are not comparable to Table 3. *The typed row's $41/200$ is Table 3's dependent chain, downstream leaf recalibrated per split: a different object from the rows below it. The ungated row is the 7B leaf's own error with no gate, not the two-stage OR loss.

method	draws	returns	risk bound	leaf labels
λ-Cert (typed composition)*	0	$41/200$ draws	0.100	700/leaf
end-to-end resampling, n=200	200	always	0.031	–
end-to-end resampling, n=500	500	always	0.021	–
end-to-end resampling, n=1000	1000	always	0.017	–
end-to-end resampling, n=2000	2000	always	0.014	–
ungated 7B leaf (reference point)	0	–	0.070 observed	–

Table 12: RQ3 on one object. The 3B→7B pipeline with both thresholds frozen at 0.76, a loss if either stage errs, risk $\Pr[\text{accept} \wedge L]$; every bound is computed on the evaluation fold (n=300) of each of 200 splits and its coverage is judged on the 700 held-out questions neither bound read, at $\delta_{\text{tot}} = \delta = 0.10$. The typed row reads its two leaf pools; the exact end-to-end row reads the fold's own labels once; the resampled rows redraw those labels with replacement, and their coverage falls below $1-\delta$ from the first row. No row generates or labels anything new.

certificate	bound (median [5–95%])	coverage	labels drawn from	new labels
typed, α_{top} at $\delta_{\text{tot}} = 0.10$	0.100	1.000	700 cached (calibration fold)	0
CP on own n=300, $\delta = 0.10$	0.022 [0.008–0.031]	0.950	300 cached (evaluation fold)	0
resampled, n=200 draws	0.026 [0.011–0.046]	0.894	200 draws of 300 cached	0
resampled, n=500 draws	0.018 [0.005–0.033]	0.791	500 draws of 300 cached	0
resampled, n=1000 draws	0.015 [0.002–0.027]	0.732	1000 draws of 300 cached	0
resampled, n=2000 draws	0.013 [0.001–0.024]	0.670	2000 draws of 300 cached	0

Table 13: RQ4, in full; Figure 5a plots the violation rates. The budget correction against its absence, at α=δ=0.05 over 200 splits. Left: grid size K on the 7B GSM8K gate, one fold selecting among K thresholds. Right: k stages sharing calibration items. Violation and FWER are out-of-sample and should stay at or below 0.05. Coverage is at four decimals and K=11 is shown, so each coverage gap the text quotes is a difference of two entries here.

	corrected		uncorrected			corr.	uncorr.
grid sweep	violation	coverage	violation	coverage	stage sweep	FWER	FWER
grid K=5	0.015	0.7967	0.015	0.9033	k=1 stage	0.000	0.000
grid K=11	0.015	0.8667	0.020	0.9033	k=2 stages	0.005	0.005
grid K=26	0.015	0.8633	0.020	0.9033	k=3 stages	0.020	0.020
grid K=51	0.015	0.8633	0.020	0.9033			
grid K=101	0.015	0.8567	0.020	0.9033			
grid K=201	0.015	0.8567	0.020	0.9033			