WHEN MORE CAUTION CAN MAKE LLMS LESS SAFE: RISK CONTROL FOR ITERATIVE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Iterative LLM systems emit candidate answers over several rounds and must decide when to act, continue, or escalate; given a budget β, we seek a stopping threshold controlling the wrong-action rate $R = \Pr(\text{act} \wedge \text{wrong})$, which is what "safer" means here. Greater caution can increase this risk, since a higher threshold may delay action until a later round overturns an already-correct answer, so the pointwise loss monotonicity conformal risk control (CRC) requires can fail; decomposing the risk change between nested rules says exactly when caution helps. More strongly, the full risk and automation curves, the score-process distribution and every round's accuracy do not reveal whether that premise holds: two populations can agree on all of them, yet in one no trajectory violates pointwise monotonicity and in the other all do. They differ only in how correctness at one round is coupled to correctness at another, so no non-trivial bound on the violating fraction follows from any of those curves: a failure of identification, not of estimation, which an oracle would not resolve. Empirically the difficulty grows with reasoning: on MMLU-Pro, using prefixes of one $T = 8$ run, accuracy gains saturate while correct answers broken climb from 1.9% to 8.7% of instances and the share of calibration draws with an upward risk step rises from 33% to 90%, while the aggregate curve remains nearly flat. We therefore certify stopping policies directly, keeping calibration labels isolated from score fitting, grid and budget, which gives $\Pr(R(\hat{\lambda}) \leq \beta) \geq 1 - \gamma$ without monotonicity and without any assumption on the score. Iterative systems need trajectory-aware certification, because aggregate risk cannot reveal whether trajectory-level monotonicity holds.

1 INTRODUCTION

Multi-agent debate, in which several LLM instances exchange and revise answers over rounds, improves accuracy on reasoning tasks (Du et al., 2024; Liang et al., 2024). Putting such a system into an autonomous pipeline raises a question accuracy numbers do not answer: for a given instance, after a given number of rounds, should the system act on the current answer, deliberate further, or escalate out of the pipeline? The quantity an operator can be held to is the rate of *wrong actions*, meaning instances the system acted on autonomously and got wrong. We study the problem in the form deployment poses it: given a declared wrong-action budget β, produce a stopping rule that provably respects it while retaining as much automation as certification permits.

That is *selective autonomy*: on each instance the system may commit its current answer as the autonomous output, continue reasoning, or leave the instance unresolved and escalate it for human or tool-based review. Escalation terminates the autonomous process, so the cost and accuracy of whatever handles an escalated instance stay outside the guarantee, which is why expert-aware learning-to-defer asks a different question, optimising a system loss that includes the downstream expert (Madras et al., 2018; Mozannar & Sontag, 2020; Verma & Nalisnick, 2022; Okati et al., 2021). Nor is the setting specific to debate: it needs only a predictor that revises its answer over time while emitting a score, which covers self-refinement, repeated sampling and verifier-guided loops. We state the theory at that generality and use deliberation as the primary empirical setting because it makes iterative revision explicit, confirming in Section 5.1 that single-agent self-revision shows the same phenomenon.

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.

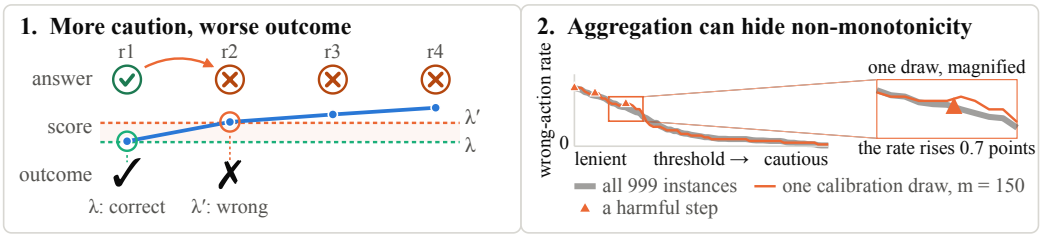

Figure 1: **More caution can make iterative reasoning less safe, and the aggregate curve need not show it.** *Panel 1*: one real trajectory, MMLU-Pro instance 2104, correct at round 1 and overturned at round 2 while confidence climbs $0.22 \to 0.62 \to 0.77 \to 0.91$, so the lenient λ acts on the correct round and the cautious λ' on the broken one. This is the break configuration (B) of Theorem 1; 229 instances across the six benchmarks share its shape. *Panel 2*: the wrong-action rate against the threshold on MMLU-Pro, over all 999 instances (grey) and one calibration draw of $m = 150$ (orange). The draw steps up where the pooled curve does not: what an operator sees depends on the draw, and neither view certifies the premise. Both panels are diagnostics, not guarantees.

Split conformal prediction (Vovk et al., 2005; Angelopoulos & Bates, 2023) and conformal risk control (CRC) (Bates et al., 2021; Angelopoulos et al., 2024) look applicable, and they run into *two independent obstacles*; only the first is specific to iterative reasoning. **Structural:** CRC needs each instance's own loss to be monotone in the threshold, multi-round stopping can violate that, and Section 3 shows the premise cannot be certified from the risk curve even if that curve were known exactly. **Statistical:** the remedies for non-monotone losses (Angelopoulos et al., 2024; Angelopoulos, 2026; Aldirawi et al., 2026) are valid and control what they target, the *expected* risk over calibration draws, which is not what an operator who calibrates once needs. Fixed-sequence Learn-then-Test avoids both at once; neither obstacle implies the other, and Section 5.3 separates them.

Existing work has established risk-controlled stopping for LLM reasoning: Wang et al. (2026b) calibrate a reasoning model's token budget by distribution-free risk control, and Wang et al. (2026a) act when a split-conformal set over deliberating agents is a singleton. We ask a different question, about the object being calibrated rather than the calibrator. When a system may *revise* its answer, wrong-action risk need not be monotone in the threshold, so demanding more confidence can make the deployment less safe, and no risk curve, however exactly known, can certify the pointwise monotonicity raw CRC requires. The first of those works thresholds a confidence signal along a reasoning trajectory and inherits this directly; the second is exposed differently, and Section 6 says how. Figure 1 shows the mechanism and what aggregation does to it, on the quantities Section 2 defines.

Contributions. **(i) Mechanism.** An exact decomposition of the risk difference between two thresholds into a right-to-wrong mass, a wrong-to-right mass, and the wrong actions a higher threshold defers away (Theorem 1). Caution is safe exactly when the first does not exceed the other two, so wrong-action risk need not be monotone in the threshold and raw CRC's premise can fail. Every term is the probability of an observable event, so the decomposition doubles as a diagnostic. **(ii) Identification.** Even exact risk and automation curves only partially identify the harmful right-to-wrong mass (Proposition 3); and the risk curve, the automation curve, the score-process law and every round's accuracy can all be identical under zero versus complete violation of the premise, and under zero versus complete correctness churn (Theorem 4, Corollary 5), with three rounds enough and two not. What those marginals cannot see is the coupling across rounds: a failure of identification, not of estimation. **(iii) Certification.** The calibrator is not the contribution: we instantiate existing fixed-sequence Learn-then-Test (Angelopoulos et al., 2025) inside a leakage-free four-way split that keeps score fitting, grid construction and budget estimation off the calibration labels. That gives $\Pr\left(R(\hat{\lambda}) \le \beta\right) \ge 1 - \gamma$ with no assumption on the score and none beyond i.i.d. sampling (Theorem 6). **(iv) Evidence.** Reading every horizon off one $T = 8$ run per setting as a prefix, what extra rounds add saturates while what they break does not: on MMLU-Pro, correct answers broken climb from 1.9% to 8.7% of instances and the share of calibration draws with an

upward risk step from 33% to 90%, while the aggregate curve remains nearly flat. The mechanism is not debate-specific: it appears with a single revising agent and in iterative code self-revision, where correctness is decided by executing tests the model never sees.

2 RISK-CONTROLLED SELECTIVE AUTONOMY

We call a predictor *iterative* if, on an input x, it emits a sequence of candidate answers $\hat{Y}_1, \ldots, \hat{Y}_T$ together with a transcript $\mathcal{F}_t$ that records everything observable through round t. Nothing below uses more structure than this, so the theory applies to any iterative predictor of this form; our experiments instantiate two, several agents deliberating and one agent revising itself.

Concretely, for a multiple-choice task with inputs $\mathcal{X}$ and labels $\mathcal{Y} = \{1, \ldots, C\}$, N agents ($N \geq 1$) run for at most T rounds and $\mathcal{F}_t$ holds every agent's response and elicited option scores. We average the round-t scores uniformly into a *social belief* $P^{(t)}_{\text{social}}(y \mid x)$ and take $\hat{Y}_t = \arg\max_y P^{(t)}_{\text{social}}(y \mid x)$, which for $N = 1$ is just that model's own belief; Appendix G gives elicitation and tie-breaking as implemented.

Policies. A *score* is any function $S_t = s(\mathcal{F}_t) \in \mathbb{R}$ computed from the transcript, possibly using a frozen auxiliary dataset. Given a threshold λ, the induced policy π_λ stops at

$$\tau(\lambda) \;=\; \min\{t \leq T : S_t \geq \lambda\}, \qquad \min \emptyset := T + 1, \tag{1}$$

acts on $\hat{Y}_{\tau(\lambda)}$ if $\tau(\lambda) \leq T$, and otherwise *escalates* the instance out of the autonomous pipeline. The baseline "always act at round T" is *not* in this family, since first crossing makes $\lambda \to -\infty$ act at round 1; we include it separately as a round-T reference in the experiments. A score need not be a calibrated probability of correctness, and nothing below assumes it is: its only role is to induce an ordered family of policies, and *score* always means that generic ordering statistic, of which *confidence* is one instance. That ordering fixes what *caution* means here: $\tau(\lambda)$ is non-decreasing in λ, so a larger threshold never stops earlier, and that is the only sense in which one policy is more cautious than another. The quantities of interest are the *wrong-action risk*, the *automation rate*, and the *acted-on accuracy*:

$$R(\lambda) = \Pr\big(\tau(\lambda) \leq T,\ \hat{Y}_{\tau(\lambda)} \neq Y\big), \quad A(\lambda) = \Pr\big(\tau(\lambda) \leq T\big), \quad \mathrm{Acc}|\mathrm{Act} = 1 - R(\lambda)/A(\lambda). \tag{2}$$

Throughout, *safer* means a lower R, not a broader notion of end-to-end system safety, and R is a joint probability, not a conditional one, because that is the quantity a budget is naturally stated in. Minimising R alone is trivial, since escalating everything achieves $R = 0$, so the deployment problem is: given β, return a policy with $R \leq \beta$ and A as large as possible. We reserve R, A and $\mathrm{Acc}|\mathrm{Act}$ for population quantities; WA, Act and the acted-on accuracy always denote their empirical held-out estimates, and we keep the two vocabularies apart because an empirical curve does not tell you a population one.

The only sampling assumption for the risk-control guarantee.

Assumption 1 (i.i.d. sampling)**.** *The tuples (X_i, Y_i, ξ_i), where ξ_i collects all randomness the agents use on instance i, are i.i.d. and independent of the test instance (X, Y, ξ).*

This is stronger than the exchangeability usually invoked in conformal prediction, and deliberately so: our argument treats per-instance wrong-action indicators as independent Bernoulli draws, which exchangeability alone does not give, since an exchangeable sequence may share a latent variable. Nothing else is assumed for Theorem 6: in particular no smoothness of any conditional-correctness function, no sufficiency of any state representation, and no property whatsoever of the score. The optional conditional analysis of Appendix E adds regularity assumptions of its own, which buy a different statement and are never used here.

When is a budget binding? If β exceeds the population final-round error $e_T = \Pr(\hat{Y}_T \neq Y)$, then "always act at round T" is feasible at 100% automation and no stopping rule can beat it, so an evaluation in that regime measures nothing, which is easy to do by accident: a budget at or above the final-round error, $\kappa \geq 1$ in the notation below, cannot bind. We therefore report absolute budgets and budgets $\beta = \kappa\, \hat{e}_T$ with $\kappa < 1$, and always show the always-act row so the reader can see whether the

constraint bites. As e_T is unknown, the pre-LTT estimate $\hat{e}_T$ indicates only whether a budget is likely to bind and is never a certificate: we do not claim always-act provably exceeds the budget, only that it does so on every benchmark and split at every budget below $\hat{e}_T$.

3 How stopping risk moves with the threshold

Conformal risk control calibrates λ by inverting an empirical risk curve, and requires monotonicity: in its usual form, that each instance's own loss, its *pointwise* loss, is monotone in λ, which is stronger than monotonicity of the population risk R (Angelopoulos et al., 2024). For single-shot selective prediction it holds, since raising the threshold only removes instances from the acted-on set. Multi-round stopping is different, and the following identity says how. Write $E_t = \mathbf{1}[\hat{Y}_t \neq Y]$ for the round-t error indicator, and call τ, τ' *nested* if $\tau \leq \tau'$ pointwise; nothing below uses the threshold form of (1), or indeed anything about how the answers were produced, and two thresholds $\lambda < \lambda'$ are nested because the acting set shrinks, $\{t : S_t \geq \lambda'\} \subseteq \{t : S_t \geq \lambda\}$.

Theorem 1 (Risk-transition decomposition)**.** *For any nested stopping rules $\tau \leq \tau'$, writing $R(\tau) = \Pr(\tau \leq T, \hat{Y}_\tau \neq Y)$,*

$$R(\tau') - R(\tau) \;=\; \underbrace{\Pr\left(\tau' \leq T,\, E_\tau = 0,\, E_{\tau'} = 1\right)}_{B\ (\textit{break})\colon\ \textit{right} \to \textit{wrong}} \;-\; \underbrace{\Pr\left(\tau' \leq T,\, E_\tau = 1,\, E_{\tau'} = 0\right)}_{F\ (\textit{fix})\colon\ \textit{wrong} \to \textit{right}} \;-\; \underbrace{\Pr\left(\tau \leq T < \tau',\, E_\tau = 1\right)}_{D\ (\textit{defer})\colon\ \textit{wrong action deferred away}}. \tag{3}$$

The proof (Appendix A) partitions on $\{\tau' \leq T\}$, $\{\tau \leq T < \tau'\}$ and $\{\tau > T\}$ and cancels the events where the two rounds agree; thresholds are the special case $\tau = \tau(\lambda),\ \tau' = \tau(\lambda')$. Each term is the probability of an observable event, making (3) an empirical diagnostic as well as a proof device: Section 5.2 evaluates all three terms on every adjacent pair of the grid, recovering the identity to numerical precision. Stopping later helps through F and D, by waiting for a round that repairs a wrong answer and by declining to act at all on some instances it would have got wrong, and hurts through B, by waiting past a round that was already right; it is safe exactly when $B \leq F + D$.

Corollary 2 (Non-monotonicity)**.** *Writing $Z_t = 1 - E_t$ for the correctness indicator, there is a distribution over $(S_1, \ldots, S_T, Z_1, \ldots, Z_T)$ and thresholds $\lambda < \lambda'$ with $R(\lambda) < R(\lambda')$.*

All proofs are in Appendix A; the corollary follows by putting a point mass on $S_1 = 0.6,\ S_2 = 0.9$ with $Z_1 = 1,\ Z_2 = 0$, which zeroes F and D and sets $B = 1$. What such a construction needs is not a non-monotone score, since the score rises here, but a *correctness transition* whose score crosses λ' only after the transition: exactly what conformity pressure (Zhang et al., 2024) and social reinforcement toward a shared mistake with rising stated confidence (Wang et al., 2026a) produce. Whether B exceeds $F + D$ is empirical, and Section 5.2 finds "rarely, but not never".

One might hope the premise is checkable, since the risk curve one plots usually looks monotone. It is not, and it fails twice over. Write $L(\lambda) = \mathbf{1}[\tau(\lambda) \leq T,\ E_{\tau(\lambda)} = 1]$ for one instance's own loss, so that $R(\lambda) = \mathbb{E}\, L(\lambda)$. What CRC needs is that $L(\cdot)$ is non-increasing for *each* instance; what one can see is the average. The first failure is quantitative. Suppose an operator knows two operating points exactly: how often each of two policies acts, and how often each is wrong. How many individual answers could the extra caution have broken?

Proposition 3 (Sharp partial identification of the harmful mass)**.** *Let $\lambda < \lambda'$ and write $r = R(\lambda)$, $r' = R(\lambda')$, $a = A(\lambda)$. Then*

$$\max\{0,\, r' - r\} \;\leq\; B \;\leq\; \min\{r',\, a - r\},$$

and every value in that interval is attained by a population with those same three observables.

Only to within an interval, then: two exact operating points do not say how many answers the extra caution broke. The second failure is total and needs the whole curve rather than two points on it: two rounds already suffice for non-monotone risk (Corollary 2), but three are needed for the stronger phenomenon below, complete non-identifiability of trajectory-level monotonicity from the entire curve.

Theorem 4 (Complete non-identifiability of pointwise monotonicity)**.** *Let $T \geq 3$. There is a single non-increasing risk function $R^\star$ and trajectory distributions P_η, one for each $\eta \in [0,1]$, with*

$$R_{P_\eta}(\lambda) = R^\star(\lambda) \quad \textit{for} \text{ every } \lambda \in \mathbb{R} \qquad \textit{and} \qquad P_\eta\big(L(\cdot) \text{ non-monotone}\big) = \eta.$$

So L is non-increasing P_0-almost surely while every *trajectory of P_1 is non-monotone. Moreover the P_η agree on much more than the risk curve: they share the law of $S_{1:T}$, hence $A(\cdot)$,* and *every round's error probability* $\Pr(E_t = 1)$*. All that separates them is how correctness at one round is coupled to correctness at another, and three rounds are the fewest for which such a family exists (Lemma 9).*

Corollary 5 (No non-trivial aggregate certificate)**.** *Let $T \geq 3$, write $V(P) = P\big(L(\cdot) \text{ non-monotone}\big)$, and let the* correctness churn $C(\lambda, \lambda') = \Pr(\tau' \leq T,\, E_\tau \neq E_{\tau'}) = B + F$ *be the chance the two policies act on answers of differing correctness.* (i) *No map Ψ from a risk function $R(\cdot)$ to $\{0,1\}$ reporting whether L is non-increasing almost surely is correct on every trajectory distribution.* (ii) *Call $\Psi : R(\cdot) \to [0,1]$ a* valid certificate *if $\Psi(R_P) \geq V(P)$ for every such P. Then $\Psi(R^\star) = 1$: on a risk function realised by a population whose loss is monotone almost surely, the only valid certificate is the vacuous one.* (iii) *The same P_0, P_1 admit thresholds $\lambda < \lambda'$ at which they agree on R and on A, yet $C_{P_0}(\lambda, \lambda') = 0$ while $C_{P_1}(\lambda, \lambda') = 1$: the curve records the net flow $B - F - D$ and is blind to the churn $B + F$ behind it.*

All three parts are immediate: P_0 and P_1 share a risk function but differ in the property, so a binary Ψ is wrong on one of them, and a valid certificate must cover $V(P_1) = 1$ while $V(P_0) = 0$. Part (ii) bites: an operator who would settle for "at most 1% of trajectories violate the premise" cannot have that from the curve either. Labelled trajectories do carry the information; the curve does not, and not because of discretisation: the populations agree at *every* threshold, so an oracle for the population risk is no better off. Section 4 therefore calibrates without the premise instead of checking it.

4 Calibration without monotonicity

Fixed-sequence testing addresses both obstacles with one certification step: it requires no monotonicity assumption and provides high-probability control over the calibration draw rather than control only in expectation.

4.1 Fixed-sequence Learn-then-Test

Certifying candidate stopping rules directly takes three steps. **Separate.** Split the training half four ways: $\mathcal{D}_{\text{mod}}$ and $\mathcal{D}_{\text{loc}}$ fit the score, $\mathcal{D}_{\text{grid}}$ fixes the candidate thresholds Λ, a relative budget β is estimated from those three splits together, and $\mathcal{D}_{\text{ltt}}$ (size m) is used only for certification, so score, grid and budget are all fixed before a calibration label is touched, which is the measurability Theorem 6 asks for. **Order.** $\Lambda = \{\lambda_1 > \cdots > \lambda_M\}$ is the set of empirical quantiles of the score on $\mathcal{D}_{\text{grid}}$, taken in decreasing order, so the sequence starts at the most conservative candidate and walks towards more automation (Appendix C explains what goes wrong if the grid is read off $\mathcal{D}_{\text{ltt}}$ instead). **Test and stop.** Test each candidate for being over budget and stop at the first one that cannot be certified safe. We refer to this four-way-split fixed-sequence procedure as the *LTT wrapper*: it returns one certified member of a pre-specified family of stopping rules without altering the underlying score.

The test is exact, and because the loss here is a $0/1$ indicator it needs no concentration bound. For each grid point the null is that the budget is violated, $H_j : R(\lambda_j) > \beta$, and the statistic is the integer wrong-action count on $\mathcal{D}_{\text{ltt}}$,

$$W_j = \sum_{i \in \mathcal{D}_{\text{ltt}}} \mathbf{1}[\tau_i(\lambda_j) \leq T,\, \hat{Y}_{i,\tau_i} \neq Y_i],$$

and the exact binomial p-value

$$p_j \;=\; \Pr\big(\text{Bin}(m, \beta) \leq W_j\big). \tag{4}$$

The tail direction is worth a sentence, because it looks inverted at first glance: the null is that the policy is *unsafe*, so it is an unusually *small* number of observed wrong actions that counts as evidence against it, hence the lower tail. Under H_j, W_j stochastically dominates $\text{Bin}(m, \beta)$, so p_j is super-uniform. This needs no monotonicity and no property of the score, only that the indicators are i.i.d. Bernoulli

given the frozen score and grid, which is Assumption 1. The rest is the standard fixed-sequence test: walk the grid down, stop at the first j with $p_j > \gamma$, deploy the highest-automation candidate rejected before that point, and defer everything if none was (Algorithm 1, Appendix A). The test is existing machinery (Angelopoulos et al., 2025); the separation above is what makes it valid here.

Theorem 6 (Validity without monotonicity)**.** *Let Assumption 1 hold, let Λ and β be measurable with respect to $\mathcal{D}_{\mathrm{mod}} \cup \mathcal{D}_{\mathrm{loc}} \cup \mathcal{D}_{\mathrm{grid}}$, and let $\hat{\lambda}$ be the output of Algorithm 1. Write $R(\lambda)$ for the wrong-action risk of π_λ on a fresh instance, conditional on those three splits, so that the score is a fixed function. Then*

$$\Pr\left(R(\hat{\lambda}) \le \beta\right) \;\ge\; 1 - \gamma, \tag{5}$$

where the probability is over the draw of $\mathcal{D}_{\mathrm{ltt}}$ and the event $\{\hat{\lambda} = \textsc{none}\}$ (defer everything, $R = 0$) lies inside the good event. No assumption is made about the score, and R need not be monotone in λ.

The argument (Appendix A) turns on the first grid index whose true risk exceeds β: the sequence cannot pass it without falsely rejecting it, an event of probability at most γ. Learn-then-Test recasts calibration as multiple testing, so any procedure controlling the family-wise error rate yields a valid selection; we use the fixed sequence because it avoids a multiplicity penalty, which matters at the calibration sizes the small benchmarks afford ($m = 70$ on GPQA). To select from the calibration data in some other way, use the simultaneous variant:

Theorem 7 (Simultaneous validity)**.** *Under the conditions of Theorem 6, let $\hat{\Lambda} = \{\lambda_j \in \Lambda : p_j \le \gamma/M\}$. Then $\Pr\left(\forall \lambda \in \hat{\Lambda} : R(\lambda) \le \beta\right) \ge 1 - \gamma$, so any selection rule with output in $\hat{\Lambda}$ inherits the guarantee.*

Corollary 8 (Acted-on error)**.** *On the event of* (5)*, if $A(\hat{\lambda}) > 0$ then $\Pr\left(\hat{Y}_\tau \ne Y \mid \tau \le T\right) \le \min\{1, \beta/A(\hat{\lambda})\}$.*

So a budget can be met by acting rarely and well or often and worse; only the joint quantity R is controlled.

Remark 1 (Choosing among scores costs γ)**.** *Each row of our score comparison is a separate policy with its own $(1-\gamma)$ guarantee. Selecting among J scores on the same $\mathcal{D}_{\mathrm{ltt}}$ costs γ/J per score, or select on $\mathcal{D}_{\mathrm{grid}}$; the comparison we report is not a valid selection procedure and we do not use it as one.*

Two different targets. The contrast with CRC is not only about assumptions but about what is promised:

$$\underbrace{\mathbb{E}_{\mathcal{D}_{\mathrm{ltt}}}\left[R(\hat{\lambda})\right] \le \beta}_{\text{expected-risk target (CRC)}} \qquad \text{versus} \qquad \underbrace{\Pr_{\mathcal{D}_{\mathrm{ltt}}}\left(R(\hat{\lambda}) \le \beta\right) \ge 1 - \gamma}_{\text{high-probability target (used here)}}. \tag{6}$$

We calibrate once, so we target the second: the first averages over calibration draws, and the draw in hand can still be one whose risk exceeds β. We call the second *draw-wise* control. CRC inverts the empirical risk and is tighter when applicable, but needs the monotone loss Section 3 shows this problem lacks, and its remedies keep the left-hand target; split conformal targets marginal coverage, a different functional, and thresholds tuned on a training split have no finite-sample guarantee at all.

4.2 Scores affect utility, not validity

Theorem 6 makes validity score-agnostic, so score design affects only utility: among policies carrying the same guarantee, a better score is one that automates more. We compare 4: the top social probability $p_t^{(1)}$, an isotonic recalibration of it, a gradient-boosted correctness predictor, and a local k-nearest-neighbour lower confidence bound on state-conditional correctness, the natural choice for anyone wanting a state-conditional reliability estimate. Appendices D–F define them and explain why we claim no validity for the last one's empirical modulus. We compare them under identical certification in Section 5.3.

5 Experiments

Setup. Three heterogeneous agents (Claude Haiku 4.5 (Anthropic, 2025), DeepSeek-R1 (Guo et al., 2025) and Qwen3-32B (Yang et al., 2025)) deliberate for $T = 4$ rounds, each round conditioning

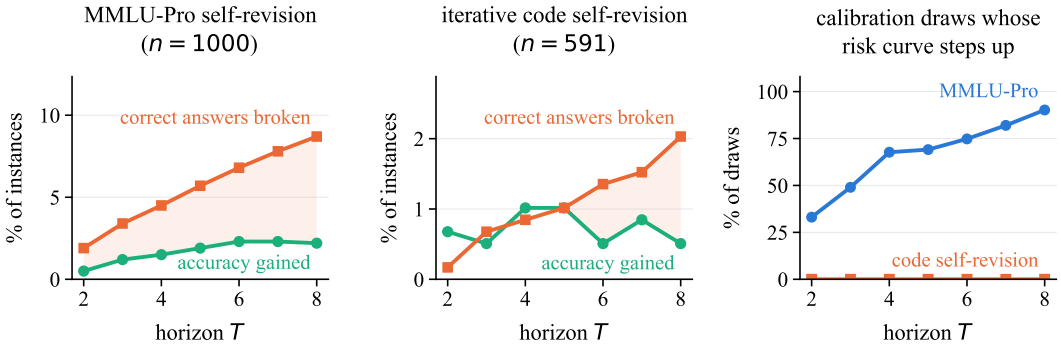

Figure 2: **More reasoning improves accuracy while breaking more correct answers.** Every horizon is a *prefix* of one $T = 8$ run per setting, so instances, model and sampling are identical across T. The first two panels share one axis: the gain saturates while the breakage climbs, overtaking it in code self-revision. The third asks how often a practitioner calibrating from a finite sample would see that instability: the fraction of 1,000 calibration sets of size 150 whose empirical risk curve contains any upward step, 33% rising to 90% on MMLU-Pro and 0% at every horizon for code self-revision (exact definition in Appendix L.1, per-horizon values in Table 17).

on the others' previous responses and confidence distributions (Appendix G), on six benchmarks: MMLU-Pro (Wang et al., 2024), LogiQA (Liu et al., 2020), ARC-Challenge (Clark et al., 2018), BIG-Bench Hard (Suzgun et al., 2023), MuSR (Sprague et al., 2024) and GPQA (Rein et al., 2024), with sizes in Table 3. Debates are run once per instance and cached, so every result below is re-derivable without further LLM calls. Each dataset is split in half into training and test over 10 random splits, and the training half is split again into four disjoint parts: $\mathcal{D}_{\text{mod}}$ and $\mathcal{D}_{\text{loc}}$ to fit the score, $\mathcal{D}_{\text{grid}}$ to place the grid, $\mathcal{D}_{\text{ltt}}$ to certify. That separation is not an implementation preference but the hypothesis of Theorem 6, which needs Λ and β fixed without seeing $\mathcal{D}_{\text{ltt}}$. We use $\gamma = 0.10$ and report automation and wrong-action rates on the held-out test halves, each with its own $\hat{\lambda}$, so the numbers describe the procedure rather than one deployed policy. Split proportions, the score's hyperparameters and how the test halves are aggregated are in Appendix G.

Section 5.1 needs $T = 8$, so it adds two cached runs, both *single-agent self-revision* with Claude Haiku 4.5: MMLU-Pro at $T = 8$ scored by the same elicited distribution ($n = 1000$), and *iterative code self-revision* on the union of HumanEval and the MBPP test split ($n = 591$), which asks whether the phenomenon survives outside multiple choice. There each round returns a complete function and a verbalized confidence, and the model revises only its own previous program: it never sees the benchmark tests or their output, which we execute afterwards to decide correctness (Appendix L.1).

5.1 The phenomenon: longer reasoning breaks more correct answers

Does more reasoning make the stopping problem harder, or is the phenomenon an accident of four rounds? Using prefixes of a single $T = 8$ run per setting lets us track accuracy gain, answer breakage and risk-curve non-monotonicity on exactly the same trajectories (Figure 2). What the extra rounds add saturates ($+0.5$ to $+2.2$ points), while the fraction *broken* (a correct answer followed by a wrong one at a later adjacent round, repaired later or not) climbs from 1.9% to 8.7% of instances, and so does the share of calibration draws with an upward step, though the full-set curve's largest upward step stays at most 0.10pp.

Outside multiple choice the separation is sharper still: in code self-revision, broken answers rise from 0.2% to 2.0% while accuracy moves by $+0.5$ points, yet none of the sampled calibration curves, nor the full-set curve, shows an upward step at any horizon. This is not a contradiction: a right-to-wrong revision changes the threshold-wise loss only when the stopping rule can act on opposite sides of that revision. So the experiment shows the narrower claim, that answer instability can grow while the observed risk curve stays flat (Appendix L.2).

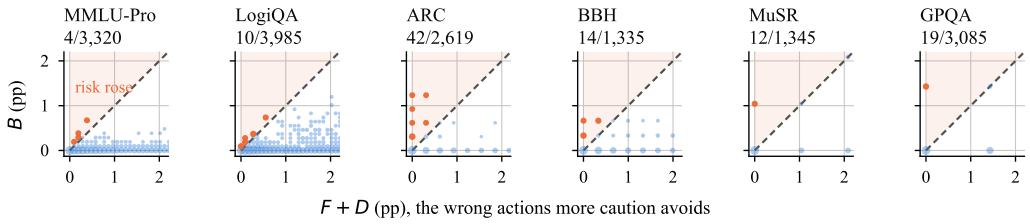

Figure 3: **More caution raises the risk exactly when B exceeds $F + D$, and that happens on every benchmark.** Threshold pairs over the 10 splits and six separations; on the dashed line Theorem 1 makes $R(\lambda') = R(\lambda)$. Orange points in the shaded half-plane raised the risk: we call such a pair *harmful*, meaning $R(\lambda') > R(\lambda)$ or equivalently $B > F + D$. Each subtitle counts harmful pairs over all pairs, at most 1.6%, and never zero. Both masses are multiples of one over the calibration-split size $|\mathcal{D}_{\text{ltt}}|$, so pairs coincide on the smaller benchmarks: marker area is how many sit at a point. Pair totals per benchmark are in Table 15, Appendix J.

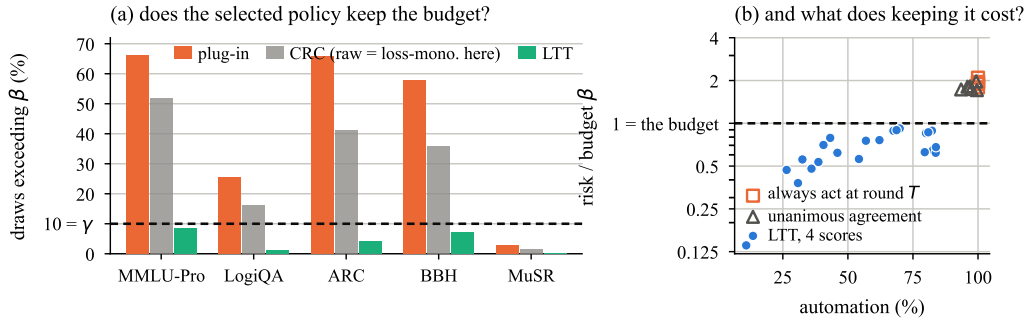

Figure 4: **Testing the budget controls the deployment risk, with an explicit automation trade-off. (a)** How often one calibration returns a policy whose risk on a deployment set no selector saw exceeds β, per benchmark, at $m = 150$ over 400 draws. The target is draw-wise, which is not CRC's own guarantee. LTT's observed exceedance is at or below the nominal γ on all 5, the others on all but MuSR, whose loose budget every selector keeps. The CRC bar is both the published procedure and the loss-monotonised variant of Angelopoulos et al. (2024), which coincide here and remain valid for their expected-risk target. **(b)** The price, in units of each benchmark's own budget, on a log axis; each blue point is one score–benchmark pair. All 4 scores stay under budget, differing only in automation, 26–84%; the references break it at near-full automation (Appendix I.3).

Peer disagreement is not necessary. A matched 707-question comparison against two three-agent protocols, at fixed elicitation and generation budget, shows every protocol breaking correct answers: self-revision at 3.5% while its accuracy rises +1.7 points, the disjoint triple highest at 5.2%. Completion is instance-dependent, so the ordering is descriptive (Appendix N).

5.2 The mechanism: when the broken answers outweigh the repaired ones

We first evaluate all three terms of Theorem 1 on every adjacent grid pair over the same 10 splits; the identity holds to 10^{-16}, and summed over pairs B is 6.6 to 15.4 times smaller than D, which is why raising the threshold is usually safe. Adjacent quantiles move little mass, so Figure 3 plots the same decomposition at six separations, 1 to 99 grid steps, the boundary $B = F + D$ being the same for any nested pair. Pairs above it occur on 3–7 of the 10 splits per benchmark, the largest increase 4.17pp on MuSR. Such pairs are rare, but the guarantee asks for absence, not rarity, and no aggregate curve certifies absence: being about correctness transitions, the precondition cannot be settled from the score process or from unlabelled data, and even both curves exactly known leave Proposition 3's interval, 86–166 times the harmful mass it contains (Appendix J). The decomposition also suggests

policies that *model* these transitions, and the LTT wrapper certifies any pre-specified family of them (Corollary 11); Appendix K.2 builds four and none reliably beats a calibrated scalar.

5.3 Certification: testing the budget instead of assuming monotonicity

Since the premise cannot be certified from the curve, we test the budget directly, with β set to half the final-round error estimated on each split's pre-LTT portion, which binds on all six benchmarks. Figure 4(b) is the result: the uncalibrated policies exceed it everywhere, the LTT-certified rules keep their pooled held-out wrong-action rate below it on all six, and automation follows reliability, 83.8% on BBH against 26.5% on MuSR: a budget buys automation only where correctness is predictable. Table 3 in Appendix I.2 gives automation, wrong-action rate and per-split exceedance counts for every cell; executed rounds and acted-on accuracy are in Appendix I.8 and the supplement.

More elaborate scores bring no consistent gain: plain calibrated $p_t^{(1)}$ wins 16 of the 31 binding cells (Appendix I.3). What first crossing buys over stopping at round T is latency, not automation: 15–54% fewer rounds at essentially the same automation (Appendix I.8).

What draw-wise control buys. Does the distinction between expected-risk and draw-wise control matter in practice? Figure 4(a) says yes. Nothing here depends on non-monotonicity: this is the second obstacle. It needs a reservoir larger than m and a disjoint deployment set, which GPQA at $n = 546$ cannot afford, so it covers 5 benchmarks. We add a *deployment set* no selector sees, redraw the calibration set 400 times, and evaluate each selected threshold there, so only the selection rule differs.

Figure 4(a) shows the separation: LTT is at or below its nominal 10% on 5 of 5 benchmarks, reaching 8.5% at most, against 66.2% for the plug-in selector and 51.7% for the CRC variants. This is not CRC failing at its own task, since the monotonised variant controls the expected risk it targets; it is the gap between the two targets of (6), measured. The LTT wrapper pays for the stronger target in automation, and the ordering holds in all 18 cells (Table 6, Appendix I).

The obstacles stay distinct: these exceedances come from finite-sample optimism, since almost none of the selected thresholds sat past a harmful step. Appendix I.5 collects the aggregation effect with the search-order questions and a test of Theorem 6 against a closed-form population risk.

6 Related work

Debate improves reasoning and a growing line studies its stability (Du et al., 2024; Liang et al., 2024; Qian et al., 2025; Choi et al., 2025; Zhang et al., 2024; Li et al., 2024; Xie et al., 2025; Hu et al., 2025; Yang et al., 2024); the break mass B of Theorem 1 is that instability's decision-theoretic consequence. Conformal prediction and CRC calibrate a threshold against a monotone loss (Vovk et al., 2005; Angelopoulos & Bates, 2023; Bates et al., 2021; Angelopoulos et al., 2024), and two recent works keep the expected-risk target under non-monotone losses (Angelopoulos, 2026; Aldirawi et al., 2026): the left of (6), where we take the right. We use existing calibration machinery; our contributions are to identify iterative answer revision as a setting where the monotonicity premise can fail, characterise when it fails, and show why aggregate risk cannot certify it. Appendix B covers Wang et al. (2026a) and the neighbouring literatures.

7 Conclusion

Additional reasoning can improve average accuracy while making safe stopping structurally harder, and that difficulty is invisible in the aggregate risk curve even when it is known exactly: marginal summaries cannot certify trajectory-level behaviour in iterative systems. What is missing is the cross-round coupling of correctness, which labelled trajectories expose. Nothing here is specific to language models: wherever a later stage can overturn an earlier stage's correct decision, the same upward risk steps arise and the same decomposition locates them. **Limitations.** The guarantee is over the calibration draw: not instance-conditional, shift-robust or per subgroup (Appendix O); escalation is free; our horizon is eight rounds. Relaxing the first is where we would go next.

Reproducibility statement

The paper has two layers and they are reproducible to different degrees. **Analysis.** Every number, table and figure is computed by a script in the supplement from the released per-round arrays; none is typed in by hand. The bundle pins its dependencies and carries the split seeds, protocol settings and the aligned question identifiers the matched comparisons need, and it ships a verification script that recomputes the tables and fails if any of them differs from the released copy. **Generation.** What we release are those derived arrays (for every instance and round, the answer, whether it was correct, and the score) rather than the raw transcripts, which are too large to ship and in one case no longer exist; the prompts, request metadata and a digest for every run are in the supplement, as is each run we exclude and why. Temperature and decoding seeds were never pinned, so re-running generation does not return the same transcripts: that step is documented rather than replayable, which is why the released arrays, not the model, are the starting point for every reported number. See Section 5, Appendices G and N, and the supplement README for the full path.

AI use statement

The authors conceived and carried out the research, including the theoretical development, methodology, experimental design, analysis, and manuscript preparation. Generative AI tools were used as supporting tools for literature discovery, coding and analysis assistance, and language editing. LLMs were also used to generate the deliberation transcripts analysed in the experiments. The authors verified all technical content and take full responsibility for the paper.

References

Tareq Aldirawi, Yun Li, and Wenge Guo. Conformal risk control under non-monotone losses: Theory and finite-sample guarantees. *arXiv preprint arXiv:2604.01502*, 2026.

Anastasios N Angelopoulos. Conformal risk control for non-monotonic losses. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 384(2327): 20250068, 2026. doi: 10.1098/rsta.2025.0068.

Anastasios N Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

Anastasios N Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. *International Conference on Learning Representations*, 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. *The Annals of Applied Statistics*, 19(2):1641–1662, 2025. doi: 10.1214/24-AOAS1998.

Anthropic. Introducing Claude Haiku 4.5. `https://www.anthropic.com/news/claude-haiku-4-5`, 2025. Model announcement, 15 October 2025.

Rina Foygel Barber, Emmanuel J Candès, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021.

Rina Foygel Barber, Emmanuel J Candès, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. ChatEval: Towards better LLM-based evaluators through multi-agent debate. In *International Conference on Learning Representations*, 2024.

Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for nearest neighbor classification. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Hyeong Kyu Choi, Xiaojin Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? In *Advances in Neural Information Processing Systems*, pp. 112802–112834, 2025.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763, 2024.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *Advances in Neural Information Processing Systems*, 30, 2017.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Leying Guan. Localized conformal prediction: A generalized inference framework for conformal prediction. *Biometrika*, 110(1):33–50, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Rohan Hore and Rina Foygel Barber. Conformal prediction with local weights: randomization enables robust guarantees. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 87 (2):549–578, 2025. doi: 10.1093/jrsssb/qkae103.

Tianyu Hu, Zhen Tan, Song Wang, Huaizhi Qu, and Tianlong Chen. Multi-agent debate for LLM judges with adaptive stability detection. In *Advances in Neural Information Processing Systems*, pp. 51926–51962, 2025.

Geoffrey Irving, Paul Christiano, and Dario Amodei. AI safety via debate. *arXiv preprint arXiv:1805.00899*, 2018.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R Bowman, Tim Rocktäschel, and Ethan Perez. Debating with more persuasive LLMs leads to more truthful answers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23662–23733, 2024.

Bhawesh Kumar, Charlie Lu, Gauri Gupta, Anil Palepu, David Bellamy, Ramesh Raskar, and Andrew Beam. Conformal prediction with large language models for multi-choice question answering. *arXiv preprint arXiv:2305.18404*, 2023.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. More agents is all you need. *Transactions on Machine Learning Research*, 2024.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17889–17904, 2024.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. LogiQA: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3622–3628, 2020.

David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7076–7087, 2020.

Nastaran Okati, Abir De, and Manuel Gomez-Rodriguez. Differentiable learning under triage. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Scaling large language model-based multi-agent collaboration. In *International Conference on Learning Representations*, 2025.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S Jaakkola, and Regina Barzilay. Conformal language modeling. In *International Conference on Learning Representations*, 2024.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. GPQA: A graduate-level Google-proof Q&A benchmark. In *First Conference on Language Modeling (COLM)*, 2024.

Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In *Conference on Robot Learning (CoRL)*, 2023.

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *International Conference on Learning Representations*, 2024.

Charles J Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5(4):595–620, 1977.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

Alexandre B Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. A stitch in time saves nine: Detecting and mitigating hallucinations of LLMs by validating low-confidence generation. *arXiv preprint arXiv:2307.03987*, 2023.

Rajeev Verma and Eric Nalisnick. Calibrated learning to defer with one-vs-all classifiers. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22184–22202, 2022.

Vladimir Vovk. Conditional validity of inductive conformal predictors. *Machine Learning*, 92(2–3): 349–376, 2013.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Mengdie Flora Wang, Haochen Xie, Guanghui Wang, Aijing Gao, Guang Yang, Ziyuan Li, Qucy Wei Qiu, Fangwei Han, Hengzhi Qiu, Yajing Huang, Bing Zhu, and Jae Oh Woo. From debate to decision: Conformal social choice for safe multi-agent deliberation. *arXiv preprint arXiv:2604.07667*, 2026a.

Sijia Wang and Lifu Huang. Debate as optimization: Adaptive conformal prediction and diverse retrieval for event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16422–16435, 2024.

Xi Wang, Anushri Suresh, Alvin Zhang, Rishi More, William Jurayj, Mehrdad Farajtabar, Daniel Khashabi, and Eric Nalisnick. Conformal thinking: Risk control for reasoning on a compute budget. In *International Conference on Machine Learning*, 2026b.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, pp. 95266–95290, 2024.

Yi Xie, Zhanke Zhou, Chentao Cao, Qiyu Niu, Tongliang Liu, and Bo Han. From debate to equilibrium: Belief-driven multi-agent LLM reasoning via Bayesian Nash equilibrium. In *International Conference on Machine Learning*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Joshua C Yang, Damian Dailisan, Marcin Korecki, Carina I Hausladen, and Dirk Helbing. LLM voting: Human choices and AI collective decision-making. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, pp. 1696–1708, 2024. doi: 10.1609/aies.v7i1.31758.

Jintian Zhang, Xin Xu, Ningyu Zhang, Ruibo Liu, Bryan Hooi, and Shumin Deng. Exploring collaboration mechanisms for LLM agents: A social psychology view. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 14544–14607, 2024.

A Proofs for Sections 3 and 4

Proof of Theorem 1. Nestedness gives $\tau \le \tau'$ pointwise, hence $\{\tau' \le T\} \subseteq \{\tau \le T\}$; for two thresholds $\lambda < \lambda'$ this holds because $\{t : S_t \ge \lambda'\} \subseteq \{t : S_t \ge \lambda\}$. Partition the sample space into $\Omega_1 = \{\tau' \le T\}$ (both act), $\Omega_2 = \{\tau \le T < \tau'\}$ (only the earlier rule acts) and $\Omega_3 = \{\tau > T\}$ (neither acts). On Ω_3 neither policy acts, so it contributes nothing to either risk. Hence

$$R(\tau') = \Pr(\Omega_1,\, E_{\tau'} = 1), \qquad R(\tau) = \Pr(\Omega_1,\, E_\tau = 1) + \Pr(\Omega_2,\, E_\tau = 1),$$

the second because on Ω_2 the earlier rule acts at τ while the later one does not act at all. Subtracting,

$$R(\tau') - R(\tau) = \big[\Pr(\Omega_1, E_{\tau'} = 1) - \Pr(\Omega_1, E_\tau = 1)\big] - \Pr(\Omega_2, E_\tau = 1).$$

Splitting the bracket on the value of E_τ and noting that the events with $E_\tau = E_{\tau'} = 1$ appear in both terms and cancel,

$$\Pr(\Omega_1, E_{\tau'} = 1) - \Pr(\Omega_1, E_\tau = 1) = \Pr(\Omega_1, E_\tau = 0, E_{\tau'} = 1) - \Pr(\Omega_1, E_\tau = 1, E_{\tau'} = 0),$$

which are B (a right action broken) and F (a wrong action fixed); the remaining term is D (a wrong action deferred away). No property of the score enters, so the identity applies to any nested pair of stopping rules, learned or thresholded. □

Proof of Corollary 2. Take $T = 2$ and a point mass on $S_1 = 0.6$, $S_2 = 0.9$ with $Z_1 = 1$, $Z_2 = 0$: the ensemble is right at round 1 and talked out of it by round 2 while its score rises. With $\lambda = 0.5$ and $\lambda' = 0.8$ we get $\tau = 1$ and $\tau' = 2$, so $E_\tau = 0$, $E_{\tau'} = 1$, and $B = 1$ while $F = D = 0$. By Theorem 1, $R(\lambda') - R(\lambda) = 1 > 0$. □

Proof of Proposition 3 and Theorem 4. Throughout, an instance is a trajectory $(S_1, \dots, S_T, Z_1, \dots, Z_T)$, $\tau(\lambda) = \min\{t : S_t \ge \lambda\}$ with $\min \emptyset := T + 1$, and $L(\lambda) = \mathbf{1}[\tau(\lambda) \le T, E_{\tau(\lambda)} = 1]$. Since $\tau(\cdot)$ is non-decreasing in λ, any two thresholds $\lambda < \lambda'$ are nested in the sense of Theorem 1. Note first that $L(\cdot)$ is non-increasing if and only if the instance has no right-to-wrong step, i.e. contributes 0 to B for every such pair: L can only increase by passing from acting correctly to acting wrongly.

Proposition 3. Fix any $\lambda < \lambda'$; nothing here needs them to be adjacent on a grid, and write $r = R(\lambda)$, $r' = R(\lambda')$, $a = A(\lambda)$.

Lower bounds. $B \geq 0$ trivially, and Theorem 1 gives $B = (r' - r) + F + D \geq r' - r$ because $F, D \geq 0$.

Upper bounds. The event behind B is $\{\tau' \leq T,\ E_\tau = 0,\ E_{\tau'} = 1\}$. Dropping the first correctness condition gives $B \leq \Pr(\tau' \leq T, E_{\tau'} = 1) = r'$. Dropping the second instead, and using $\tau \leq \tau' \leq T$ so that E_τ is defined, gives $B \leq \Pr(\tau \leq T, E_\tau = 0) = A(\lambda) - R(\lambda) = a - r$, the mass that acts correctly at the lenient threshold. Neither step uses monotonicity.

The bound involves $A(\lambda)$ and not $A(\lambda')$: since $r' \leq A(\lambda') \leq A(\lambda)$, the later automation rate is implied by the earlier one here and tightens neither end.

The interval is non-empty. $r \leq a$ and $r' \leq A(\lambda') \leq A(\lambda) = a$ because A is non-increasing, so $\max\{0, r' - r\} \leq \min\{r', a - r\}$.

Attainment. Fix any feasible (r, r', a) and any t in the interval. Take $T = 2$ and six instance types. Four carry $S = (\lambda, \lambda')$, so they act at round 1 under λ and at round 2 under λ'; label them by (E_1, E_2) and give them masses

$$p_{01} = t, \qquad p_{11} = r' - t, \qquad p_{10} = r - r' + t, \qquad p_{00} = a - r - t.$$

One carries $S = (\lambda, \lambda)$, so it acts at round 1 under λ and defers under λ'; give it mass 0. The last never acts and takes the remaining $1 - a$. Non-negativity of the four masses is exactly $t \leq r'$, $t \geq r' - r$, $t \leq a - r$ and $t \geq 0$, i.e. exactly membership in the interval of Proposition 3. Reading the definitions off the table, $A(\lambda) = a$, $R(\lambda) = p_{10} + p_{11} = r$, $R(\lambda') = p_{01} + p_{11} = r'$, $B = p_{01} = t$, $F = p_{10}$ and $D = 0$, so the three observables are as prescribed and B takes the chosen value. Putting the mass p_{10} on the deferring type instead moves it from F to D and leaves r, r', a and B unchanged, which is how the churn of Corollary 5(iii) is varied at fixed observables.

Theorem 4. The construction is on the whole threshold line, not on a grid, and uses three rounds. Fix $a < b < M$ and give *every* trajectory below the same score vector $S = (a, b, M)$, so that only correctness varies. Then $\tau(\lambda) = t$ exactly when λ lies in the t-th of the intervals

$$I_1 = (-\infty, a], \qquad I_2 = (a, b], \qquad I_3 = (b, M],$$

and no trajectory acts for $\lambda > M$. Consequently the loss path of a trajectory is just its error indicators read off those intervals, $L = (E_1, E_2, E_3)$ on I_1, I_2, I_3 and $L = 0$ above M, so choosing $Z = 1 - E$ chooses the path. Take the four trajectories

$$\begin{array}{lll} E = (0,0,0): & (0,0,0) & \text{non-increasing,} \\ E = (1,1,1): & (1,1,1) & \text{non-increasing,} \\ E = (0,1,0): & (0,1,0) & \text{rises at } a, \quad \text{non-monotone,} \\ E = (1,0,1): & (1,0,1) & \text{rises at } b, \quad \text{non-monotone,} \end{array}$$

all read on I_1, I_2, I_3 and all 0 above M, and set

$$P_0 = \tfrac{1}{2}\,\delta_{(0,0,0)} + \tfrac{1}{2}\,\delta_{(1,1,1)}, \qquad P_1 = \tfrac{1}{2}\,\delta_{(0,1,0)} + \tfrac{1}{2}\,\delta_{(1,0,1)}.$$

Both pairs average to $\frac{1}{2}$ in each of the three intervals, since $(0,0,0) + (1,1,1) = (1,1,1) = (0,1,0) + (1,0,1)$, and both sides vanish above M. Hence

$$R_{P_0}(\lambda) = R_{P_1}(\lambda) = \tfrac{1}{2} \text{ for } \lambda \leq M, \qquad R_{P_0}(\lambda) = R_{P_1}(\lambda) = 0 \text{ for } \lambda > M,$$

one non-increasing function, flat over the whole acting range: a practitioner plotting it sees nothing unusual at all. Every path in P_0's support is non-increasing and every path in P_1's support is non-monotone, so

$$V(P_0) = 0, \qquad V(P_1) = 1.$$

Because the two laws induce the *same* risk function, so does every mixture: for $\eta \in [0, 1]$ the law $P_\eta = \eta P_1 + (1 - \eta) P_0$ has risk function R_{P_0} and $V(P_\eta) = \eta$. A single risk function is therefore shared by populations realising every achievable amount of non-monotonicity, 0 and 1 included. For $T > 3$ append rounds with score M and the round-3 correctness, which changes no stopping time and no loss.

What the family holds fixed is worth spelling out, because it is more than the theorem statement of a weaker version would give. All four trajectories carry the same deterministic scores, so the law

of $S_{1:T}$, and with it $A(\cdot)$, is common to every P_η. Each round's error probability is also common: $\Pr(E_t = 1) = \frac{1}{2}$ for $t = 1, 2, 3$ under both P_0 and P_1, since each is an equal mixture of a vector and its complement. So an observer given the exact risk curve, the exact automation curve, the exact score-process law and the exact accuracy of every individual round still cannot tell P_0 from P_1. Every one of those quantities is a *marginal*: a statement about one round, or about one threshold, at a time. What separates the two populations is only the *coupling* across rounds, whether the instance that is right at round 1 is the one that is wrong at round 2. That is the general form of the paper's message, and the reason a certificate cannot be read off any collection of marginal summaries.

□

Lemma 9 (Two rounds are not enough)**.** *Let $T = 2$. If every trajectory of P has a non-monotone loss path then R_P is not non-increasing, while $R_{P'}$ is non-increasing whenever $L(\cdot)$ is non-increasing P'-almost surely. No such pair shares a risk function, so the family of Theorem 4 does not exist at $T = 2$ and the three rounds it uses are the fewest possible.*

Proof. Let $T = 2$ and consider a single trajectory with scores (s_1, s_2). If $s_2 \le s_1$ then no $\lambda > s_1$ is met by either round, so $L(\lambda) = E_1 \mathbf{1}[\lambda \le s_1]$, which is non-increasing. If $s_1 < s_2$ then $L = (E_1, E_2, 0)$ on $(-\infty, s_1]$, $(s_1, s_2]$ and (s_2, ∞), which rises only in the case $E_1 = 0$, $E_2 = 1$. So the non-monotone trajectories at $T = 2$ are exactly those with $s_1 < s_2$, $E_1 = 0$ and $E_2 = 1$: right at round 1, wrong at round 2, on a rising score.

Suppose now $V(P) = 1$, so P-almost every trajectory is of that form. Then $R_P(\lambda) = \Pr(s_1 < \lambda \le s_2)$, because the round-1 action is never wrong and the round-2 action always is. By dominated convergence $R_P(\lambda) \to 0$ as $\lambda \to -\infty$; note that we cannot instead evaluate R_P below the essential infimum of s_1, which may be $-\infty$. Were R_P non-increasing, that left limit would force $R_P \equiv 0$, since a non-negative non-increasing function with limit 0 at $-\infty$ vanishes identically. But

$$\int_{\mathbb{R}} R_P(\lambda)\,\mathrm{d}\lambda = \mathbb{E}\big[(s_2 - s_1)^+\big] > 0$$

because $s_1 < s_2$ almost surely, a contradiction. So R_P is not non-increasing. On the other hand, if $L(\cdot)$ is non-increasing P'-almost surely then $R_{P'}$, an average of non-increasing functions, is non-increasing. The two cannot coincide, so at $T = 2$ no pair with $V = 0$ and $V = 1$ shares a risk function and the family of Theorem 4 does not exist. Three rounds are therefore necessary as well as sufficient. The reason is structural: at $T = 2$ a non-monotone path is forced to be $(0, 1, 0)$, so demanding that every trajectory violate monotonicity also fixes every trajectory's correctness pattern, and with it the risk curve. A third round is the first that admits two *different* non-monotone patterns, $(0, 1, 0)$ and $(1, 0, 1)$, whose mixture can imitate the average of two monotone ones. □

Proof of Corollary 5. Let $R^\star$ be the common risk function of the pair just constructed. For part (ii), a valid certificate must satisfy $\Psi(R^\star) \ge V(P_1) = 1$, so $\Psi(R^\star) = 1$; and $R_{P_0} = R^\star$ as functions, so Ψ receives the same argument on P_0 and returns the same number, namely 1, while $V(P_0) = 0$. The only valid certificate at $R^\star$ is thus the vacuous one, on a risk function realised by a population whose loss is monotone almost surely. Part (i) follows: a binary Ψ must answer the same way on P_0 and P_1 and is wrong on one of them. Nothing here is specific to V: the argument applies to any functional of the trajectory law that differs between P_0 and P_1, so the risk function supports no non-trivial inference about any of them.

For part (iii), take $\lambda \le a$ and $\lambda' \in (a, b]$, so that $\tau(\lambda) = 1$ and $\tau(\lambda') = 2$ for every trajectory in either population. Under P_1, the atom $(0, 1, 0)$ has $E_1 = 0, E_2 = 1$ and the atom $(1, 0, 1)$ has $E_1 = 1, E_2 = 0$, so each contributes in a different direction: $B = F = \frac{1}{2}$, $D = 0$ and $C_{P_1} = 1$. Under P_0, $(0, 0, 0)$ has $E_1 = E_2 = 0$ and $(1, 1, 1)$ has $E_1 = E_2 = 1$, so $B = F = 0$ and $C_{P_0} = 0$. Both populations have $R(\lambda) = R(\lambda') = \frac{1}{2}$ and $A(\lambda) = A(\lambda') = 1$. The net flow $B - F - D$ is 0 in both cases, which is all the risk difference reports; the gross flow differs by everything available. □

Part (ii) is worth separating from part (i) because the two say different things about what a practitioner can do. Part (i) rules out answering a yes/no question that a cautious operator might reasonably regard as too strong to ask. Part (ii) rules out the weaker request they would actually make of the curve, a bound on how much of the population violates the premise, and rules it out completely, not merely up to some level: one and the same non-increasing risk function is realised both by a population in

which no trajectory violates pointwise monotonicity and by one in which every trajectory does. The aggregate curve is not an imperfect diagnostic that more data would sharpen; it carries no information about this structure at all. Labelled trajectories do carry it, since the non-monotone share is directly estimable from them, which is what Section 5.2 does; the impossibility is about what the curve determines, not about what is knowable.

What the corollary does *not* say is that a risk-curve-only procedure must choose a worse *threshold*. Since P_0 and P_1 induce the same risk function, a selector reading only that function returns the same λ on both and attains the same wrong-action risk on both, so it loses nothing measured in R. What it cannot do is report anything true and non-trivial about the trajectory structure underneath. Turning this into a statement about policy choice would need an objective that depends on that structure together with a construction in which the budget forces the selector into the region where the structure differs; in the construction above the deployed threshold is pointwise-unsafe for half the population throughout $(a, c]$ and safe outside it, and we do not claim such a forcing argument.

Proofs for the calibration procedure. Throughout we condition on $\mathcal{D}_{\mathrm{mod}} \cup \mathcal{D}_{\mathrm{loc}} \cup \mathcal{D}_{\mathrm{grid}}$, so that the score s, the grid Λ and the budget β are all fixed; every probability below is over the draw of $\mathcal{D}_{\mathrm{ltt}}$ and of the test instance. This conditioning is what the four-way split of Section 4 buys, and Appendix C explains why it cannot be replaced by conditioning on the $\mathcal{D}_{\mathrm{ltt}}$ scores. Write $W_i(\lambda) = \mathbf{1}[\tau_i(\lambda) \le T,\ \hat{Y}_{i,\tau_i} \neq Y_i]$ for the wrong-action indicator of calibration instance i at threshold λ, and $W(\lambda) = \sum_{i \in \mathcal{D}_{\mathrm{ltt}}} W_i(\lambda)$ for the integer count, so that $R(\lambda) = \mathbb{E}[W_i(\lambda)]$.

Lemma 10 (Super-uniformity)**.** *Fix λ measurable with respect to $\mathcal{D}_{\mathrm{mod}} \cup \mathcal{D}_{\mathrm{loc}} \cup \mathcal{D}_{\mathrm{grid}}$ and let $p(\lambda) = \Pr(\mathrm{Bin}(m, \beta) \le W(\lambda))$. Under Assumption 1, if $R(\lambda) > \beta$ then $\Pr(p(\lambda) \le u) \le u$ for all $u \in [0, 1]$.*

Proof. Given the conditioning, λ and the map $\mathcal{F} \mapsto S$ are deterministic, so the variables $\{W_i(\lambda)\}_{i \in \mathcal{D}_{\mathrm{ltt}}}$ are i.i.d. Bernoulli with the common mean $R(\lambda)$: each depends only on its own instance (X_i, Y_i, ξ_i), and these are i.i.d. by Assumption 1 and independent of the splits we conditioned on. Hence $W(\lambda) \sim \mathrm{Bin}(m, R(\lambda))$. Note that the common mean is what fails if λ is instead a function of the $\mathcal{D}_{\mathrm{ltt}}$ scores; see Appendix C. If $R(\lambda) > \beta$ then $\mathrm{Bin}(m, R(\lambda))$ stochastically dominates $\mathrm{Bin}(m, \beta)$, so $\Phi_\beta(m\widehat{R}(\lambda))$ with $\Phi_\beta(x) = \Pr(\mathrm{Bin}(m, \beta) \le x)$ is stochastically larger than the value Φ_β takes on its own distribution, which is super-uniform. Concretely, for $Q \sim \mathrm{Bin}(m, \beta)$ and $Q' \sim \mathrm{Bin}(m, R(\lambda))$ with $Q \preceq Q'$, $\Pr(\Phi_\beta(Q') \le u) \le \Pr(\Phi_\beta(Q) \le u) \le u$, where the last step is the standard super-uniformity of a discrete CDF evaluated at its own variate. The argument uses the integer count; evaluating Φ_β at a floating-point reconstruction of it can round the argument down by one and destroy the inequality. This is an implementation trap and not only a formality: recovering the count from a rate as $(W_j/m) \cdot m$ can land on $6.999\ldots$, at which (4) evaluates one step too low and the test becomes anti-conservative. □

This is the only place probability enters, and it uses nothing about λ beyond it being fixed in advance. In particular monotonicity of $\lambda \mapsto R(\lambda)$ is never needed.

Proof of Theorem 6. If every $R(\lambda_j) \le \beta$ then any output is valid and the claim is trivial, as is the case $\hat{\lambda} = \textsc{none}$, for which $R = 0$. Otherwise let

$$j^\star = \min\{j : R(\lambda_j) > \beta\},$$

so that $R(\lambda_j) \le \beta$ for every $j < j^\star$. Algorithm 1 returns some $\hat{\lambda} = \lambda_{\hat{J}}$ where $\hat{J}$ is an index in the initial run of consecutive rejections. If $\hat{J} < j^\star$ then $R(\hat{\lambda}) \le \beta$ and we are on the good event. The complementary case $\hat{J} \ge j^\star$ requires the loop to have passed index $j^\star$ without breaking, i.e. to have rejected $H_{j^\star}$, i.e. $p_{j^\star} \le \gamma$. Since $R(\lambda_{j^\star}) > \beta$, Lemma 10 gives $\Pr(p_{j^\star} \le \gamma) \le \gamma$. Hence $\Pr(R(\hat{\lambda}) > \beta) \le \gamma$. □

Note where the pre-specified *order* is used: the argument needs the loop to be unable to reach an unsafe index without first testing it. Fixed-sequence testing therefore avoids a multiplicity penalty, which matters on the small benchmarks where m is a few hundred; it does not make the procedure free, since $\mathcal{D}_{\mathrm{grid}}$ costs data and the exact binomial test costs power.

Algorithm 1 Budgeted act-or-defer via fixed-sequence Learn-then-Test

Require: budget β, level γ, score s fit on $\mathcal{D}_{\mathrm{mod}} \cup \mathcal{D}_{\mathrm{loc}}$, grid Λ (descending) from $\mathcal{D}_{\mathrm{grid}}$, calibration split $\mathcal{D}_{\mathrm{ltt}}$

$\hat{\lambda} \leftarrow$ NONE
for $j = 1, \ldots, M$ **do**
  compute the integer count W_j and p_j by (4) on $\mathcal{D}_{\mathrm{ltt}}$
  if $p_j > \gamma$ **then**
    break {fixed sequence stops at the first non-rejection}
  end if
  if λ_j automates at least one instance of $\mathcal{D}_{\mathrm{ltt}}$ **then**
    $\hat{\lambda} \leftarrow \lambda_j$
  end if
end for
if $\hat{\lambda} =$ NONE **then return** DEFER-ALL
return policy $\pi_{\hat{\lambda}}$: at each round t, act on $\hat{Y}_t$ if $S_t \geq \hat{\lambda}$; defer if no round qualifies

Note also what the argument does *not* use. Neither Lemma 10 nor the $j^\star$ step refers to thresholds, to nestedness, or to the score at all: the first needs only that the wrong-action indicators are i.i.d. Bernoulli once the pre-LTT splits are fixed, and the second needs only that the candidates were put in an order before the calibration labels were read. So the guarantee extends verbatim.

Corollary 11 (Validity for any pre-specified family)**.** *Let $\pi^{(1)}, \ldots, \pi^{(M)}$ be any policies that are measurable with respect to $\mathcal{D}_{\mathrm{mod}} \cup \mathcal{D}_{\mathrm{loc}} \cup \mathcal{D}_{\mathrm{grid}}$, tested in that order by* (4) *with W_j the wrong-action count of $\pi^{(j)}$ on $\mathcal{D}_{\mathrm{ltt}}$, and let $\hat{\pi}$ be the last rejected candidate (*DEFER-ALL *if none). Then $\Pr\left(R(\hat{\pi}) \leq \beta\right) \geq 1 - \gamma$ under Assumption 1.*

Proof. Identical to the proof above with λ_j replaced by $\pi^{(j)}$: the family enters only through W_j, and $j^\star = \min\{j : R(\pi^{(j)}) > \beta\}$ is well defined for any finite list. □

This is what licenses the appendix experiments that leave the threshold family: the round-restricted policies of Section 4, the singleton-set acceptance rule of Wang et al. (2026a), the transition-aware policies of Appendix K.2, and the reordered sequences of Appendix M, whose order is learned on a pilot split disjoint from $\mathcal{D}_{\mathrm{ltt}}$ and so is still fixed before the calibration labels. Each is a finite list fixed before $\mathcal{D}_{\mathrm{ltt}}$ is read, so each inherits Corollary 11 without new theory.

Proof of Theorem 7. Let $J_{\mathrm{bad}} = \{j : R(\lambda_j) > \beta\}$. By Lemma 10 and a union bound, $\Pr(\exists j \in J_{\mathrm{bad}} : p_j \leq \gamma/M) \leq |J_{\mathrm{bad}}|\gamma/M \leq \gamma$. On the complement, $\hat{\Lambda}$ contains no bad threshold, so every $\lambda \in \hat{\Lambda}$ satisfies $R(\lambda) \leq \beta$ simultaneously; any selection rule with output in $\hat{\Lambda}$ therefore returns a valid threshold. □

Proof of Corollary 8. On the event $\{R(\hat{\lambda}) \leq \beta\}$, $\Pr(\hat{Y}_\tau \neq Y \mid \tau \leq T) = R(\hat{\lambda})/A(\hat{\lambda}) \leq \beta/A(\hat{\lambda})$. □

Remark 2 (Group-conditional extension)**.** *Let $G : \mathcal{X} \to \{1, \ldots, J\}$ be a group function. Running Algorithm 1 separately within each group, on that group's share of $\mathcal{D}_{\mathrm{ltt}}$ and at level γ/J, yields $\Pr\left(\forall g : \; R_g(\hat{\lambda}_g) \leq \beta_g\right) \geq 1 - \gamma$ where $R_g(\lambda) = \Pr(\text{act} \wedge \text{wrong} \mid G = g)$, by applying Theorem 6 within each group and a union bound. Unlike the conditional local theory of Appendix E, this needs no additional smoothness or representation assumptions, though it still needs i.i.d. sampling, a group function fixed in advance, and calibration support within each group beyond those already made; the price is paid in statistical power, since each group tests with roughly m/J instances at level γ/J. Appendix O shows that this is not a cosmetic point: a pooled threshold does* not *control subgroup risk.*

B Extended related work

Four further lines of work set the context for the paper's framing and for the conditional theory of Appendix E, and are separated here from the main comparison in Section 6 because none of them calibrates a stopping rule.

The closest prior work is Wang et al. (2026a), who act when a split-conformal prediction set over deliberating agents is a singleton at a fixed round. That guarantee is marginal coverage of the set, which does not bound $\Pr(\text{act} \wedge \text{wrong})$, and because their acceptance region is a function of $(p_t^{(1)}, p_t^{(2)})$ rather than a threshold on one score, we calibrate it as an ordered family across rounds in order to compare it at our guarantee.

Conformal methods for language models certify a single decision point (Kumar et al., 2023; Quach et al., 2024), and selective generation (Ren et al., 2023; Varshney et al., 2023) works within one generation trajectory; neither varies a stopping time, so neither meets the structure this paper is about. Debate has also been applied to evaluation (Chan et al., 2024; Khan et al., 2024), information extraction (Wang & Huang, 2024) and safety (Irving et al., 2018).

Selective classification (El-Yaniv & Wiener, 2010; Geifman & El-Yaniv, 2017) supplies the accuracy–coverage framing this paper's automation and wrong-action axes inherit: a predictor may abstain, and the question is how much coverage a target error rate buys. The difference is that abstention there is a single decision on a single prediction, so the loss is monotone in the abstention threshold by construction; it is the multi-round structure, not the abstention option, that breaks monotonicity here.

Conditional coverage is known to be hard. Exact conditional validity is unattainable without strong assumptions (Vovk, 2013; Barber et al., 2021), which is why our guarantee is over the calibration draw rather than per instance, and why the local score of Appendix E is offered as a heuristic that improves the acted-on set rather than as a conditional certificate. Appendix F states explicitly which assumption we decline to make and what it would have bought.

Learning-to-defer models a human expert inside a joint system loss and optimises the two together (Madras et al., 2018; Mozannar & Sontag, 2020; Verma & Nalisnick, 2022; Okati et al., 2021). We do not: escalation is treated as available and its cost is not modelled, so our wrong-action budget is a constraint on the autonomous branch alone. Coupling the two would change the objective from "control $\Pr(\text{act} \wedge \text{wrong})$" to "minimise total system loss", and the decomposition of Theorem 1 would then need a term for the deferral cost. We regard that as the natural extension and note it in the limitations.

C On the threshold grid

The grid must be fixed before the labels of $\mathcal{D}_{\text{ltt}}$ are used. We take it to be 100 empirical quantiles of the score on $\mathcal{D}_{\text{grid}}$, a split disjoint from $\mathcal{D}_{\text{ltt}}$, so that conditional on $\mathcal{D}_{\text{mod}} \cup \mathcal{D}_{\text{loc}} \cup \mathcal{D}_{\text{grid}}$ the grid is a deterministic finite set and Lemma 10 applies to each of its points.

A tempting alternative is to place the grid at quantiles of the score on $\mathcal{D}_{\text{ltt}}$ itself and argue that conditioning on those scores is harmless. It is not. Conditional on the calibration scores $\{S_i\}_{i \in \mathcal{D}_{\text{ltt}}}$, the wrong-action indicators are independent but no longer identically distributed: $W_i(\lambda)$ is Bernoulli with mean $q_\lambda(S_i)$, so $\sum_i W_i(\lambda)$ is Poisson-binomial with mean $\sum_i q_\lambda(S_i)$ rather than $\text{Bin}(m, R(\lambda))$, and the p-value of (4) loses its calibration. Worse, the family of hypotheses being tested becomes random, so the fixed-sequence argument no longer controls anything. Both problems disappear once the grid comes from an independent split.

A data-independent grid removes the issue without any split, but is a poor substitute. With 100 equally spaced thresholds on $[0, 1]$ and otherwise identical settings, automation moves by well under a point on MMLU-Pro, LogiQA and BBH but collapses on ARC, whose scores concentrate in a narrow band near 1 because its base accuracy is around 96%: an equally spaced grid places almost no candidates where the decision lies. A practitioner who prefers a fixed grid should choose its spacing with the score distribution in mind, and the cost of getting that wrong is far larger than the cost of one extra split.

Table 1: **The smallest budget a calibration split can certify.** With no wrong action on $\mathcal{D}_{\mathrm{ltt}}$ the exact test of (4) returns $p = (1-\beta)^m$, so rejection needs $(1-\beta)^m \le \gamma$ and no budget below $\beta_{\min} = 1-\gamma^{1/m}$ is certifiable at any score: Algorithm 1 can only return DEFER-ALL. The last column is the largest wrong-action count that still rejects at $\beta = 0.05$, and it is 0 on GPQA, where certifying that budget needs a perfect calibration split. Conservative behaviour on the small benchmarks is therefore partly a statement about power rather than about the score.

Dataset	$m = \lvert\mathcal{D}_{\mathrm{ltt}}\rvert$	$\beta_{\min}$ (%)	max W at $\beta = 0.05$
MMLU-Pro	1040	0.22	42
LogiQA	1087	0.21	44
ARC	324	0.71	10
BBH	301	0.76	9
MuSR	96	2.37	1
GPQA	70	3.24	0

The budget must also come from off the calibration split. For the relative budgets $\beta = \kappa\,\hat{e}_T$, the final-round error is estimated on $\mathcal{D}_{\mathrm{mod}} \cup \mathcal{D}_{\mathrm{loc}} \cup \mathcal{D}_{\mathrm{grid}}$ and never on $\mathcal{D}_{\mathrm{ltt}}$. Estimating it on the whole training half would let the labels used in the binomial test also set the threshold the test compares against; Theorem 6 requires β to be fixed before $\mathcal{D}_{\mathrm{ltt}}$ is seen, which the split scheme above delivers.

C.1 What budget the calibration size can reach

One consequence of the exact test is worth stating because it is free and often mistaken for a property of the score. Setting $W = 0$ in (4) gives $p = (1-\beta)^m$, so a candidate can be rejected at all only if $(1-\beta)^m \le \gamma$, that is

$$\beta \;\ge\; \beta_{\min} \;=\; 1-\gamma^{1/m}.$$

Below $\beta_{\min}$ no threshold is certifiable however good the score, and Algorithm 1 returns DEFER-ALL by arithmetic. Table 1 tabulates it: 0.21–3.24% across our benchmarks, the loosest floor on $GPQA$, where $\beta = 0.05$ is reachable only with a calibration split containing no wrong action at all. So the conservative behaviour the small benchmarks show is in part a statement about power. The same substitution applies to the simultaneous variant of Theorem 7 with γ replaced by γ/M, which raises the floor considerably at $M = 100$; that variant is best reserved for the larger splits.

D The local k-NN score

On the database split $\mathcal{D}_{\mathrm{loc}}$ we record, for every round, the state $U_{i,t}$ and the correctness indicator $Z_{i,t}$. At test time, for each k in a family $\mathcal{K}$, let $\mathcal{N}_{t,k}$ be the k nearest calibration states, $h_{t,k}$ the k-NN radius, and $\hat{q}_{t,k} = \frac{1}{k}\sum_{i\in\mathcal{N}_{t,k}} Z_{i,t}$ the local correctness frequency. The score is

$$S_t \;=\; \max_{k\in\mathcal{K}} \Big[\hat{q}_{t,k}(U_t) \;-\; b_{t,k}(U_t) \;-\; \sqrt{\tfrac{\log(T|\mathcal{K}|/\delta_0)}{2k}}\Big], \tag{7}$$

where $b_{t,k}(u) = \hat{\omega}_t(h_{t,k}(u))$ is an empirical modulus of continuity estimated on $\mathcal{D}_{\mathrm{mod}}$ by smoothing Z with a logistic GAM and measuring how much the pilot varies within radius r. The constant δ_0 and the envelope $b_{t,k}$ are heuristic: they shape the score to prefer neighbourhoods that are both tight and well populated, and no validity claim attaches to them (Appendix F). The first ablation sets $b_{t,k} \equiv 0$; the second additionally drops the concentration term and fixes a single k, leaving the raw local correctness frequency.

The prediction that survives from the theory of Appendix E is where this score cannot work. Its certification slack shrinks at the nonparametric rate $O(n^{-1/(2+d_U)})$, so with a small database the neighbourhood radius does not shrink, the modulus term swamps the estimate, and the score degenerates towards a global average with a penalty attached. On GPQA $|\mathcal{D}_{\mathrm{loc}}| = 109$ with smallest admissible $k = 64$, and on MuSR $|\mathcal{D}_{\mathrm{loc}}| = 151$ with $k = 128$; those are the two benchmarks where Section 5 finds it losing, which is qualitatively consistent with the rate.

E CONDITIONAL THEORY FOR THE LOCAL SCORE

This appendix develops the local analysis that motivates the form of the score (7). *None of the guarantees in the main text depend on anything in this section.* We state it because (i) it explains why the score has the shape it does, and (ii) its rate result correctly predicts where the score fails empirically.

Conditional coverage is fundamentally hard (Vovk, 2013; Barber et al., 2021), which is why localised conformal methods reweight near the query (Guan, 2023; Hore & Barber, 2025) and Mondrian schemes partition the space (Vovk et al., 2005; Barber et al., 2023). What follows sits in that tradition and inherits its limitation: it is conditional on assumptions we cannot verify from data, whereas the deployment guarantee of Section 4 is not. The k-nearest-neighbour rates it relies on are classical (Stone, 1977; Chaudhuri & Dasgupta, 2014; Tsybakov, 2009).

Let $U_t = \phi_t(\mathcal{F}_t)$ be an $\mathcal{F}_t$-measurable state in a metric space $(\mathcal{U}, d)$, let $q_t(u) = \Pr(Y = \hat{Y}_t \mid U_t = u)$, and write $L_{t,k} = \hat{q}_{t,k}(U_t) - b_{t,k}(U_t) - c_k$ with $c_k = \sqrt{\log(T|\mathcal{K}|/\delta)/(2k)}$ and $L_t = \max_k L_{t,k}$. Let $\mathcal{H}_t = \sigma(\mathcal{D}_{\mathrm{cal}}, \mathcal{F}_t)$ and $\tau = \min\{t : L_t \geq 1 - \alpha\}$.

Assumption 2 (Local bias envelope)**.** *For each t and $k \in \mathcal{K}$ there is a measurable $b_{t,k} : \mathcal{U} \to [0, \infty)$ with $\left|\frac{1}{k}\sum_{i \in \mathcal{N}_{t,k}(u)} q_t(U_{i,t}) - q_t(u)\right| \leq b_{t,k}(u)$ almost surely for all $u \in \mathcal{U}$.*

Assumption 3 (One-sided action-region representation gap)**.** *There is $\epsilon_{\mathrm{act}} \geq 0$ such that for every t and every history f_t with $u = \phi_t(f_t)$ satisfying $q_t(u) \geq 1 - \alpha$, $\left[q_t(u) - \Pr(Y = \hat{Y}_t \mid \mathcal{F}_t = f_t)\right]^+ \leq \epsilon_{\mathrm{act}}$ a.s.*

Theorem 12 (Pointwise and simultaneous validity)**.** *Under Assumptions 1 and 2, for fixed (t, k) and $\delta_{t,k} \in (0, 1)$, $\Pr(L_{t,k}(U_t) > q_t(U_t)) \leq \delta_{t,k}$. Setting $\delta_{t,k} = \delta/(T|\mathcal{K}|)$ gives $\Pr(\exists t : L_t(U_t) > q_t(U_t)) \leq \delta$.*

Proof. Condition on $\Sigma_{\mathrm{loc}} = \sigma(\{U_{i,t}\}_i, U_t)$. Given Σ_{loc} the neighbourhood index set and radius are deterministic, and the labels $\{Z_{i,t}\}$ are independent with $\mathbb{E}[Z_{i,t} \mid U_{i,t}] = q_t(U_{i,t})$; here we use Assumption 1 rather than exchangeability, since an exchangeable sequence may share a latent variable and would not admit a conditional Hoeffding bound. Upper-tail Hoeffding gives $\hat{q}_{t,k} \leq \mu_{t,k} + c_k$ with conditional probability $\geq 1 - \delta_{t,k}$, where $\mu_{t,k} = \frac{1}{k}\sum_{i \in \mathcal{N}_{t,k}} q_t(U_{i,t})$. On that event $L_{t,k} \leq \mu_{t,k} - b_{t,k}(U_t) \leq q_t(U_t)$ by Assumption 2. Integrating over Σ_{loc} and union-bounding over the $T|\mathcal{K}|$ pairs gives the second claim. □

Theorem 13 (Wrong-action bound)**.** *Under Assumptions 1, 2 and 3,* $\Pr(\tau \leq T,\ \hat{Y}_\tau \neq Y) \leq \delta + \alpha + \epsilon_{\mathrm{act}}$.

Proof. Define the *per-round* good events $\mathcal{G}_t = \bigcap_{k \in \mathcal{K}} \{L_{t,k}(U_t) \leq q_t(U_t)\}$. Each $\mathcal{G}_t$ is $\mathcal{H}_t$-measurable, because q_t is a fixed (if unknown) function and U_t is $\mathcal{H}_t$-measurable; so is $\{\tau = t\}$, which is determined by $L_1, \ldots, L_t$. The per-round form is what makes the argument work: a global event $\bigcap_{t,k}\{L_{t,k} \leq q_t\}$ involves rounds after τ, is not $\mathcal{H}_\tau$-measurable, and cannot be conditioned on here. Decomposing over the disjoint events $\{\tau = t\}$,

$$\Pr(\tau \leq T, \hat{Y}_\tau \neq Y) = \sum_{t=1}^{T} \Pr(\tau = t, \hat{Y}_t \neq Y) \leq \sum_{t=1}^{T} \Pr(\tau = t, \hat{Y}_t \neq Y, \mathcal{G}_t) + \sum_{t=1}^{T} \Pr(\tau = t, \mathcal{G}_t^c).$$

The second sum is at most $\Pr(\bigcup_t \mathcal{G}_t^c) \leq \delta$ by Theorem 12, using disjointness of $\{\tau = t\}$. For the first, on $\{\tau = t\} \cap \mathcal{G}_t$ the stopping rule gives $L_t \geq 1 - \alpha$ and validity gives $q_t(U_t) \geq L_t \geq 1 - \alpha$, so Assumption 3 applies and $\Pr(\hat{Y}_t \neq Y \mid \mathcal{H}_t) \leq \alpha + \epsilon_{\mathrm{act}}$ on that event (the conditioning is legitimate because $\{\tau = t\} \cap \mathcal{G}_t \in \mathcal{H}_t$, and $\Pr(Y = \hat{Y}_t \mid \mathcal{H}_t) = \Pr(Y = \hat{Y}_t \mid \mathcal{F}_t)$ because $\mathcal{D}_{\mathrm{cal}}$ is independent of the test instance). Taking expectations, $\sum_t \Pr(\tau = t, \hat{Y}_t \neq Y, \mathcal{G}_t) \leq (\alpha + \epsilon_{\mathrm{act}}) \sum_t \Pr(\tau = t, \mathcal{G}_t) \leq \alpha + \epsilon_{\mathrm{act}}$. □

Proposition 14 (Cost of envelope failure)**.** *Suppose Assumption 2 is weakened to: there are measurable sets $\mathcal{N}_t \subset \mathcal{U}$ with $\Pr(U_t \in \mathcal{N}_t) \leq \eta_t$ such that the envelope inequality holds for every $u \notin \mathcal{N}_t$. Then $\Pr(\tau \leq T, \hat{Y}_\tau \neq Y) \leq \delta + \alpha + \epsilon_{\mathrm{act}} + \sum_t \eta_t$.*

Proof. Repeat the proof of Theorem 13 with $\mathcal{G}'_t = \mathcal{G}_t \cap \{U_t \notin \mathcal{N}_t\}$, adding $\sum_t \Pr(\tau = t, U_t \in \mathcal{N}_t) \le \sum_t \eta_t$. □

The formulation matters. Treating "Assumption 2 holds" as an event and applying the theorem restricted to it would not be valid, because conditioning on such an event need not preserve the i.i.d. structure the Hoeffding step requires. Quantifying failure through the U_t-probability of a bad region avoids that.

Theorem 15 (Oracle slack)**.** *Let $\mathcal{G}_2 = \{\forall t, k : |\hat{q}_{t,k} - \mu_{t,k}| \le c_k\}$, so $\Pr(\mathcal{G}_2) \ge 1 - 2\delta$. On $\mathcal{G}_2$, $q_t(U_t) - L_t(U_t) \le 2 \min_{k \in \mathcal{K}} [b_{t,k}(U_t) + c_k]$ for all t.*

Proof. Two-sided Hoeffding conditional on Σ_{loc} and a union bound over $T|\mathcal{K}|$ pairs give $\Pr(\mathcal{G}_2^c) \le 2\delta$. On $\mathcal{G}_2$ the lower tail gives $\hat{q}_{t,k} \ge \mu_{t,k} - c_k \ge q_t(U_t) - b_{t,k}(U_t) - c_k$, hence $L_{t,k} \ge q_t(U_t) - 2[b_{t,k}(U_t) + c_k]$; maximising over k gives the claim. □

Corollary 16 (Certification rate)**.** *Assume in addition local support, $\Pr(d(U_t, u) \le r) \ge c_0 r^{d_{\mathcal{U}}}$ for small r, and a Lipschitz envelope $b_{t,k}(u) \le \Lambda_t \, h_{t,k}(u)$. If $\mathcal{K}$ contains $k^\star \asymp n^{2/(2+d_{\mathcal{U}})}$ then, for a fixed δ, with probability at least $1 - 2\delta - Te^{-k^\star/8}$ we have $q_t - L_t = O(n^{-1/(2+d_{\mathcal{U}})})$. The statement is at fixed confidence, not an event of probability tending to one: driving $\delta \to 0$ with n puts a $\log(1/\delta_n)$ factor into c_k and so into the rate.*

Proof. A multiplicative Chernoff bound gives $h_{t,k}(u) \le (2k/(nc_0))^{1/d_{\mathcal{U}}}$ with probability $\ge 1 - e^{-k/8}$. The rate needs this at $k^\star$ only, so we apply it there and union-bound over the T rounds, which is why the failure term is $Te^{-k^\star/8}$ and not $T|\mathcal{K}|e^{-k^\star/8}$: a union bound over all of $\mathcal{K}$ contributes $T\sum_{k \in \mathcal{K}} e^{-k/8}$, which $|\mathcal{K}|e^{-k^\star/8}$ bounds only if every $k \in \mathcal{K}$ is at least $k^\star$. Intersecting with $\mathcal{G}_2$, Theorem 15 yields $q_t - L_t \le 2[\Lambda_t(2k^\star/(nc_0))^{1/d_{\mathcal{U}}} + c_{k^\star}]$, and balancing the two terms gives $k^\star \asymp n^{2/(2+d_{\mathcal{U}})}$ and the stated rate. Two points are easy to get wrong here: the bound requires the *two-sided* event $\mathcal{G}_2$, not the one-sided validity event, because an upper bound on the slack needs the Hoeffding lower tail, and it carries the factor 2 inherited from Theorem 15. □

Corollary 16 is the part of this theory that survives as a practically useful statement, because it predicts the failure we observe: the slack shrinks only as $n^{-1/(2+d_{\mathcal{U}})}$, so with n in the hundreds and $\min \mathcal{K}$ the neighbourhood cannot be local at all, and the score degenerates. Section 5 confirms this on GPQA and MuSR.

Proposition 17 (Group-conditional version)**.** *Fix a group function G and suppose Assumptions 2 and 3 hold with group-specific envelope $b_{t,k}^{(g)}$ for the group-conditional correctness $q_t^{(g)}$ and gap $\epsilon_{\mathrm{act}}^{(g)}$. Running the construction within group g at level δ_g gives $\Pr\left(\tau \le T, \hat{Y}_\tau \ne Y \mid G = g\right) \le \delta_g + \alpha + \epsilon_{\mathrm{act}}^{(g)}$.*

The statement is deliberately conditional on $G = g$: a bound on the *joint* probability $\Pr(\text{act}, \text{wrong}, G = g)$ is weaker, and is implied by the unconditional theorem, so it would say less than it appears to. Note also that group-conditional control requires group-specific instances of both assumptions, whereas the LTT route of Remark 2 requires none.

F WHY OUR EMPIRICAL MODULUS IS NOT CERTIFIED

The envelope we use in (7) is $b_{t,k}(u) = \hat{\omega}_t(h_{t,k}(u))$ with $\hat{\omega}_t(r) = \max\{|\tilde{q}_t(U_{i,t}) - \tilde{q}_t(U_{j,t})| : i, j \in \mathcal{D}_{\mathrm{mod}},\ d(U_{i,t}, U_{j,t}) \le r\}$ and $\tilde{q}_t$ a logistic-GAM pilot. Assumption 2 requires $\hat{\omega}_t(r) \ge \omega_t(r)$, the true modulus of q_t. Decomposing,

$$\omega_t(r) - \hat{\omega}_t(r) \le \underbrace{[\omega_t(r) - \omega_{\tilde{q}}(r)]}_{\text{pilot error}} + \underbrace{[\omega_{\tilde{q}}(r) - \hat{\omega}_t(r)]}_{\text{discretisation}} \le 2\|q_t - \tilde{q}_t\|_\infty + 2L_{\tilde{q}}\,\epsilon_{\mathrm{cover}},$$

so a sufficient condition for validity is $2\|q_t - \tilde{q}_t\|_\infty + 2L_{\tilde{q}}\,\epsilon_{\mathrm{cover}} \le 0$. Both terms are non-negative, so this particular sufficient condition holds only in the degenerate case where both vanish, which cannot generally be guaranteed from finite data. We stress what this does and does not show: it says that

the route we took to certify the empirical modulus cannot be completed, not that no certification is possible. A genuine impossibility result would require a separate argument, and a different envelope construction, for instance one built from a uniform confidence band for q_t rather than from a point estimate, might well admit a finite-sample guarantee. What follows for this paper is only that we do not have one, and so the main text makes no validity claim about $b_{t,k}$.

We also record which diagnostics are *not* evidence for the envelope. Binning held-out pairs by distance and reporting the fraction of smoothed pilot gaps below $\hat{\omega}_t(r)$ on the same bin is vacuous: $\hat{\omega}_t(r)$ is the maximum over exactly those gaps, so the fraction is 100% by construction. A stress test that inflates $\hat{\omega}_t$ and observes little change in automation is likewise not evidence that the envelope dominates ω_t: if the score is far from the threshold, it is insensitive to the penalty either way. An informative diagnostic would need a uniform confidence band for q_t, for instance Clopper–Pearson intervals on binned empirical correctness, compared against $\hat{\omega}_t$, which is a reasonable direction but is not what makes the guarantee of Section 4 work, and we do not rely on it.

G Deliberation protocol and features

Agents, serving and decoding. All three agents were queried through Amazon Bedrock in `us-east-1` with the `Converse` API. The Bedrock model or inference-profile identifiers were `us.anthropic.claude-haiku-4-5-20251001-v1:0` for Claude Haiku 4.5 (Anthropic, 2025), `us.deepseek.r1-v1:0` for DeepSeek-R1 (Guo et al., 2025) and `qwen.qwen3-32b-v1:0` for Qwen3-32B (Yang et al., 2025). Requests set `maxTokens` $= 4096$ on the command line, above the archived runner's CLI default of 2048; the run index records the value for the one run generated after that field was added, and the earlier runs record nothing, so the archive documents rather than evidences it for those. *Temperature and top-p were not supplied and therefore took provider defaults*, and no generation seed was passed, so the transcripts are not bit-reproducible by re-running the generation. This is why we ship the derived per-round arrays rather than asking a reader to regenerate them: every number in the paper is reproducible from those arrays, and the generation step is documented rather than replayable. The one fixed seed in the pipeline is unrelated to decoding: GPQA option order is shuffled with the question index as seed. For Qwen3-32B no explicit thinking-mode flag was set, so its default applied; where the response carried a separate `reasoningContent` block we concatenated it with the answer text before parsing. The same system prompt was sent to all three agents, DeepSeek-R1 included, although its model card advises against using one; we record the choice rather than defend it, and note that it applies uniformly across agents and rounds and so does not confound the round-to-round comparisons.

Agents in the auxiliary runs. The protocol study of Appendix N uses two further agent sets, served the same way through Amazon Bedrock in `us-east-1`. The *disjoint triple* is Llama 3.3 70B Instruct (Grattafiori et al., 2024), `us.meta.llama3-3-70b-instruct-v1:0`; Mistral Large 3, `mistral.mistral-large-3-675b-instruct`; and Qwen3-Next 80B-A3B, `qwen.qwen3-next-80b-a3b`, a variant of the family whose report is Yang et al. (2025). The *single agent* is Claude Haiku 4.5 at the identifier above, shown its own previous answer rather than a peer's. Mistral Large 3 has no technical report we could verify, so it is identified by its serving string only, as with Claude Haiku 4.5. These runs set `maxTokens` $= 4096$, again leaving temperature and top-p at provider defaults and passing no seed. They differ from the cached protocol in two respects that matter for reading Table 20: the prompt asks for a single option letter rather than a verbalized distribution unless the row says otherwise, and dollar figures are on-demand rates at the time of the runs and should be read as ratios. Prices, token counts and the exact prompts are in the supplement.

Prompting. Two prompts are in play and both are reproduced verbatim in the supplement (`prompts/`). The system prompt asks for meticulous step-by-step work in `<reasoning>` and `<answer>` tags. The user prompt asks, at every round, for a *verbalized probability distribution* over the answer options rather than a single letter, which is what supplies the option scores of the next paragraph. At rounds $t > 1$ it additionally contains all round-$(t-1)$ responses from all agents verbatim, under an instruction to weigh the other agents' arguments and confidence distributions and to revise if convinced. $T = 4$ rounds were run for every instance.

Social belief. $P^{(t)}_{\text{social}}(y \mid x)$ is the uniform average of the three agents' elicited option scores at round t, renormalised to sum to one, and $\hat{Y}_t$ is its argmax with deterministic first-maximum tie-breaking. It averages the elicited scores, not indicators of each agent's chosen answer.

Features. Six per-round features are recorded: $p^{(1)}_t$, the margin $\Delta_t = p^{(1)}_t - p^{(2)}_t$, $p^{(2)}_t$, the entropy of $P^{(t)}_{\text{social}}$, the fraction of agents whose own top answer equals $\hat{Y}_t$, and an indicator that $\hat{Y}_t = \hat{Y}_{t-1}$. The local k-NN score uses the two-dimensional state $(p^{(1)}_t, \Delta_t)$ with the Euclidean metric on the raw features, with no quantile normalisation; the learned baseline uses all six features plus the round index.

The uncalibrated baselines. Two policies in Table 3 are not members of $\{\pi_\lambda\}$ and need stating. *Always act at round T* acts on $\hat{Y}_T$ for every instance and never escalates, the standard non-selective reference. *Unanimous agreement* acts at the first round at which every agent's own top answer equals the social argmax,

$$\tau_{\text{unan}} = \min\{t \le T : \text{ all } N \text{ agents' argmax} = \hat{Y}_t\},$$

escalating the instance if no round qualifies, which is why its automation falls short of 100%. In the implementation this is the first-crossing rule applied to the fifth feature above, the fraction of agents agreeing with $\hat{Y}_t$, at a threshold of 1. Neither policy is calibrated, and neither has a budget parameter: they are there to show what a binding budget does to rules that ignore it.

Benchmarks. MMLU-Pro (Wang et al., 2024) ($n = 8312$) pools 8 domains and BIG-Bench Hard (Suzgun et al., 2023) ($n = 2395$) pools 10 subtasks; both are treated as one population for calibration, which is why Appendix O asks separately whether a pooled threshold is valid per domain. The others are LogiQA (Liu et al., 2020) ($n = 8678$), ARC-Challenge (Clark et al., 2018) ($n = 2590$), MuSR (Sprague et al., 2024) ($n = 756$) and GPQA (Rein et al., 2024) ($n = 546$), whose n leaves $m = 70$ calibration points.

Constants and split sizes. Each dataset is split $50/50$ into a training and a test half, and the training half is split $|\mathcal{D}_{\text{mod}}|/|\mathcal{D}_{\text{loc}}|/|\mathcal{D}_{\text{grid}}|/|\mathcal{D}_{\text{ltt}}| = 20/40/15/25$ percent. The remaining constants are $\delta_0 = 0.03$, $\gamma = 0.10$ and grid size 100, over 10 splits with seeds $0, \ldots, 9$. The neighbourhood family is $\mathcal{K} = \{128, 256, 512\}$ *filtered to those* $k \le |\mathcal{D}_{\text{loc}}|$, and the concentration term in (7) uses the size of the filtered family, so it differs across benchmarks: $\mathcal{K} = \{128, 256, 512\}$ on MMLU-Pro, LogiQA and ARC, $\{128, 256\}$ on BBH, $\{128\}$ on MuSR and $\{64\}$ on GPQA, where no member of the family fits and the implementation falls back to $k = \min(|\mathcal{D}_{\text{loc}}|, 64)$. The single-$k$ ablation uses $k = 256$ where $|\mathcal{D}_{\text{loc}}|$ allows and $k = 64$ on MuSR and GPQA where it does not; the per-benchmark values are printed in the caption of each per-benchmark table in the supplement (Appendix K.3). The empirical modulus is not the exact maximum of (7)'s definition but a tabulation of it: pairwise distances are sorted once, a running maximum of pilot gaps is taken, and the result is evaluated on 201 radii up to the 99th percentile of pairwise distance and linearly interpolated. Debates were run once per instance and cached; the 10 splits are re-splits of the same cached transcripts, so the total inference cost is $\sum_d n_d \cdot T \cdot N \approx 2.8 \times 10^5$ agent calls, not ten times that.

How the 10 test halves are aggregated. Act, WA and acted-on accuracy are pooled counts over the 10 test halves, never averages of per-split ratios: a ratio of sums, so a split that certifies a stingy threshold cannot be given the same weight as one that automates everything. A split on which no threshold is certifiable contributes its instances as deferrals rather than being dropped, since dropping it would report the procedure's automation conditional on the procedure having succeeded. Each split has its own $\hat{\lambda}$, so the pooled numbers describe the procedure and not one deployed policy; Theorem 6's guarantee is per split.

H Calibration size versus neighbourhood size

Table 2 varies the share of the training half given to $\mathcal{D}_{\text{ltt}}$, holding $\mathcal{D}_{\text{mod}}$ and $\mathcal{D}_{\text{grid}}$ fixed so that $\mathcal{D}_{\text{loc}}$ absorbs the remainder. Two effects trade off. A larger $\mathcal{D}_{\text{ltt}}$ makes the binomial test in (4) more powerful, which is what limits how loose a threshold can be certified at small m. A smaller $\mathcal{D}_{\text{loc}}$

Table 2: **Trading the neighbour database against the calibration split** ($\kappa = 0.5$; $|\mathcal{D}_{\mathrm{mod}}|$ and $|\mathcal{D}_{\mathrm{grid}}|$ held at their defaults, so $\mathcal{D}_{\mathrm{loc}}$ absorbs whatever $\mathcal{D}_{\mathrm{ltt}}$ does not take). A larger $\mathcal{D}_{\mathrm{ltt}}$ makes the binomial test more powerful; a smaller $\mathcal{D}_{\mathrm{loc}}$ starves the local score of neighbours.

		$\lvert\mathcal{D}_{\mathrm{ltt}}\rvert = 25\%$				$\lvert\mathcal{D}_{\mathrm{ltt}}\rvert = 40\%$				$\lvert\mathcal{D}_{\mathrm{ltt}}\rvert = 55\%$			
	Score	Act	WA	cert	ex	Act	WA	cert	ex	Act	WA	cert	ex
GPQA	Local k-NN LCB	56.9	10.73	10/10	3	19.0	5.71	7/10	0	12.9	5.46	9/10	0
	Confidence $p^{(1)}$	62.1	10.84	10/10	2	65.1	11.54	10/10	4	63.4	10.66	10/10	2
MuSR	Local k-NN LCB	26.5	8.44	10/10	0	41.0	14.55	10/10	1	8.4	2.88	3/10	0
	Confidence $p^{(1)}$	43.2	14.18	10/10	2	47.4	15.34	10/10	2	46.8	15.26	10/10	1

starves the local score of neighbours, and on these two benchmarks $\mathcal{D}_{\mathrm{loc}}$ is already at the edge of usability: at $|\mathcal{D}_{\mathrm{ltt}}| = 55\%$ the local score certifies on only $3/10$ MuSR splits. The local score's automation is consequently non-monotone in the split, while the confidence score, which needs no neighbour database, is nearly flat in it. With $n = 546$ on GPQA and $n = 756$ on MuSR neither component has enough data, and we would not deploy the local score on either.

I Uncertainty and coverage

I.1 The synthetic non-monotone process

Two checks below need a population whose risk is known rather than estimated, so that a coverage claim is compared against the truth and not against another estimate. We use the following process, with $T = 3$ and a trap mass $\varepsilon = 0.08$.

A fraction ε of instances are *traps*: they have scores $S = (0.55, 0.70, 0.95)$ and correctness $Z = (1, 0, 0)$. A trap acts correctly at round 1 when $\lambda \le 0.55$, acts *wrongly* at a later round when $0.55 < \lambda \le 0.95$, and never acts above 0.95, so its own loss rises as the threshold rises: this is the break configuration (B) of Corollary 2 placed at a known location. The remaining $1 - \varepsilon$ draw $u \sim \mathrm{Unif}[0, 1]$ and take $S_t = u$ at every round with $Z_t \sim \mathrm{Bernoulli}(u)$ independently across rounds, so u is both the score and the per-round correctness probability. Such an instance acts at round 1 exactly when $u \ge \lambda$ and is then wrong with probability $1 - u$.

For $\lambda \in [0, 1]$, both the risk and the automation are therefore available in closed form:

$$R(\lambda) = \varepsilon\, \mathbf{1}[0.55 < \lambda \le 0.95] + (1 - \varepsilon)\, \tfrac{(1-\lambda)^2}{2}, \qquad A(\lambda) = \varepsilon\, \mathbf{1}[\lambda \le 0.95] + (1 - \varepsilon)\, (1 - \lambda)^+,$$

and outside $[0, 1]$ nothing acts. R is smooth and decreasing away from the trap band and jumps up by ε at $\lambda = 0.55$, so the process places a known non-monotone jump at a known threshold and covers the risk regimes used below. It does not cover every budget: the jump makes the attainable risks $[0,\ 0.00115]$, from thresholds above the trap band, together with $[0.08115,\ 0.46]$, so a budget between those two intervals, $\beta = 0.05$ for instance, is matched by no threshold at all. That gap is deliberate, and it is why Table 13 reports the procedure as far more conservative than β at that level. The grid is fixed in advance, 100 points on $[0, 1]$, and never read off a calibration draw. Table 5 and the coverage check below both use this process.

I.2 Main results, every cell

Table 3 is the table behind Figure 4(b). The "ex" column tests Theorem 6 in neither direction: it counts test halves whose wrong-action *estimate* lands above β, which is an estimate of R rather than a test of the guarantee, and Section 5.3 measures operational exceedance on a deployment set instead.

I.3 Automation by score

Table 4 is the full score comparison summarised in Section 5.3. Every row is calibrated by the same wrapper and therefore carries the same guarantee, so automation is directly comparable and higher is better, subject to Remark 1: this is a comparison, not a selection procedure. Two regimes are visible across the columns. The primary scores are within a couple of points of each other on the

Table 3: **Testing the budget, not the score, is what makes it hold.** Budget $\beta = 0.5\,\hat{e}_T$, formed on each split from that split's own pre-LTT estimate of e_T and then averaged over splits, so it is fixed before any calibration label is read. The $\hat{e}_T$ column is a different estimate, the final-round error over the whole benchmark, so the two need not satisfy $\beta = 0.5\,\hat{e}_T$ exactly: on MuSR, the smallest benchmark, they are 0.180 and 0.172. The budget binds on all six benchmarks. Act = automation rate, WA = wrong-action rate, both pooled over the 10 test halves; "ex" counts test halves whose WA estimate lands above β, an estimate of R rather than a test of Theorem 6, since LTT selects a threshold whose true risk sits close to β. Bold marks a pooled WA above β. The two uncalibrated policies break the budget on every benchmark whose final-round error exceeds it; both certified rules have pooled held-out wrong-action rates below the budget on all six benchmarks. The LTT wrapper is score-agnostic (any trajectory score, certified on a split that saw nothing else, by fixed-sequence Learn-then-Test, an existing procedure (Angelopoulos et al., 2025)), and the two right-hand column groups differ only in which score it wraps.

			uncalibrated						*LTT wrapper*					
			Always act at T			Unanimous			Confidence $p^{(1)}$			Local k-NN LCB		
Dataset	n	$\hat{e}_T$	Act	WA	ex	Act	WA	ex	Act	WA	ex	Act	WA	ex
MMLU-Pro	8,312	0.157	100.0	**15.5**	10	97.4	**14.0**	10	82.3	7.1	1	80.9	6.9	1
LogiQA	8,678	0.242	100.0	**24.0**	10	96.9	**22.4**	10	67.5	10.9	2	68.7	11.0	1
ARC	2,590	0.045	100.0	**4.3**	10	99.6	**4.1**	10	36.0	1.2	0	30.8	0.9	0
BBH	2,395	0.083	100.0	**8.5**	10	99.3	**8.0**	10	82.7	2.6	1	83.8	2.8	1
MuSR	756	0.344	100.0	**34.0**	10	96.0	**32.5**	10	43.2	14.2	2	26.5	8.4	0
GPQA	546	0.278	100.0	**27.7**	10	93.5	**24.7**	10	62.1	10.8	2	56.9	10.7	3

Table 4: **Automation at a guaranteed wrong-action rate, by score** ($\beta = 0.5\,\hat{e}_T$). Every row is calibrated by the same procedure and therefore carries the same guarantee, so automation is directly comparable and higher is better. Superscripts give the number of splits (out of 10) on which LTT could certify any threshold; no superscript means $10/10$. Scores are close on the two largest benchmarks and separate widely on the smaller ones, where no single score wins.

Score	**MMLU-Pro**	**LogiQA**	**ARC**	**BBH**	**MuSR**	**GPQA**
Local k-NN LCB	80.9	68.7	30.8^{9}	83.8	26.5	56.9
without the bias term	81.5	70.1	0.0^{0}	85.1	14.2^{7}	51.1^{8}
raw local mean, no penalties	81.5	69.7	32.4^{8}	85.1	36.7	51.1^{8}
Social confidence $p_t^{(1)}$	82.3	67.5	36.0^{9}	82.7	43.2	62.1
Isotonic-calibrated $p_t^{(1)}$	80.8	69.5	10.9^{6}	83.7	32.5^{8}	54.2
Learned predictor (GBM)	80.0	69.9	38.7^{9}	79.5	40.6	46.0
Agent agreement	0.0^{0}	0.0^{0}	0.0^{0}	0.0^{0}	0.0^{0}	0.0^{0}

two largest benchmarks and separate widely on the smallest, where the local k-NN neighbourhood ceases to be local, as its $O(n^{-1/(2+d_{\mathcal{U}})})$ slack rate predicts and as $|\mathcal{D}_{\mathrm{loc}}|$ and $\min \mathcal{K}$ show before deployment (Appendix E); the ablations agree, with the raw local mean at a single k and no penalties matching or beating the full bound on 5 of six benchmarks. Adaptive first-crossing and restriction to the final round are within 0.6pp of each other in automation for confidence and 1.3pp for the local bound, while the adaptive rule executes 15–54% fewer rounds, so what adaptivity buys is latency, not reliability.

I.4 Which guarantee does a budget call for?

Ten overlapping splits cannot test a $1-\gamma$ statement, since each yields a different $\hat{\lambda}$ and the population risk is unknown. We therefore use the synthetic process of Appendix I.1, whose risk and automation are analytic, redrawing $\mathcal{D}_{\mathrm{ltt}}$ 600 times.

Table 5 separates two guarantees that are easy to conflate, and both from what the draws happened to show. We compare against the CRC variant that is *valid* here, CRC under the per-instance loss monotonisation of Angelopoulos et al. (2024); raw CRC and the aggregate running maximum are

diagnostics only, since standard theory covers neither under a non-monotone loss. Loss-monotonised CRC controls $\mathbb{E}[R(\hat{\lambda})]$, averaged over calibration draws; fixed-sequence LTT controls $R(\hat{\lambda})$ with probability $1-\gamma$ for the draw in hand, and both keep the promise they make. What differs is what one deployment gets: under loss-monotonised CRC 41% of draws yield a policy whose realised risk exceeds the budget against 5% under LTT, converting that into automation LTT does not reach (24.2% against 10.8%); neither correction changes the selection here, so this process does not separate them. Which target a budget calls for is a modelling decision.

Appendix I shows the same separation on the real benchmarks and two further families under the same wrapper: an LTT-calibrated extension of the singleton-set acceptance family of Wang et al. (2026a) is competitive but no better than plain confidence (82.1% against 82.3% on MMLU-Pro), and adaptive stability stopping is too coarse to certify anything on 5 of six benchmarks. Halving γ costs little where $\mathcal{D}_{\mathrm{ltt}}$ is large and much on ARC (36.0% against 9.4%).

I.5 Secondary diagnostics for the certification experiment

Three results sit beside Section 5.3 without bearing on its claim. The aggregation effect of Theorem 4 does appear on real data (Figure 1, Panel 2): 38–98% of size-150 draws contain a harmful step while the full-set curve has none on 3 of 4 runs. Imposing a monotone search order, which one might do to recover CRC's premise by fiat, costs at most 0.73pp of automation (Appendix K.1). Learning a search order from a pilot split improved automation in neither of two attempts (Appendix M), which Corollary 5 is the reason not to find surprising. And on the synthetic process of Appendix I.1, whose population risk is closed-form, LTT attains coverage 0.947 against its nominal 0.90, which is the only direct test of Theorem 6 available here.

I.6 Exceedance on a fixed deployment set

Table 6 is the main calibration size, quoted in Section 5.3; Table 7 below repeats the study at both sizes and for both scores.

Table 5: **Two guarantees, both kept, only one of them per deployment.** On the synthetic process of Appendix I, whose population risk and automation are analytic, $\mathcal{D}_{\mathrm{ltt}}$ is redrawn 600 times. The last two columns say different things. "Guarantee applies" is about theory: under the non-monotone per-instance losses of Section 3, a distribution-free guarantee at the declared β holds for LTT, which never used monotonicity, and for CRC after the per-instance loss monotonisation $\tilde{L}_i(\lambda) = \sup_{\lambda' \geq \lambda} L_i(\lambda')$ of Angelopoulos et al. (2024). It does not hold for raw CRC, for which Aldirawi et al. (2026) instead give validity at a level inflated by $O(\sqrt{\log M/m})$, nor for the running maximum of the *aggregate* risk; those two rows are diagnostics. "Target met empirically" is about these 600 draws alone, and holds everywhere. What separates the two *valid* routes is the fraction of deployments whose *realised* risk exceeds the budget: at $\beta = 0.10$, 41% under loss-monotonised CRC against 5% under LTT, at the cost of automation. Neither correction changes the selection here, so this process does not separate them.

β	Calibration	Evaluation target	$\mathbb{E}[R]$	$\Pr(R \leq \beta)$	over β	Act	Guarantee applies	Target met empirically
0.05	LTT, fixed sequence	$\Pr(R \leq \beta)$	0.001	1.000	0%	3.7	✓	✓
	LTT, Bonferroni	$\Pr(R \leq \beta)$	0.001	1.000	0%	3.7	✓	✓
	CRC	$\mathbb{E}[R]$	0.001	0.997	0%	3.8	×	✓
	CRC, loss-monotonised	$\mathbb{E}[R]$	0.001	0.997	0%	3.8	✓	✓
	CRC, risk-monotonised	$\mathbb{E}[R]$	0.001	0.997	0%	3.8	×	✓
0.10	LTT, fixed sequence	$\Pr(R \leq \beta)$	0.037	0.947	5%	10.8	✓	✓
	LTT, Bonferroni	$\Pr(R \leq \beta)$	0.003	1.000	0%	4.2	✓	✓
	CRC	$\mathbb{E}[R]$	0.090	0.587	41%	24.2	×	✓
	CRC, loss-monotonised	$\mathbb{E}[R]$	0.090	0.587	41%	24.2	✓	✓
	CRC, risk-monotonised	$\mathbb{E}[R]$	0.090	0.587	41%	24.2	×	✓

Table 6: **The plug-in selector and both CRC variants exceed a per-deployment budget on many calibration draws; fixed-sequence LTT's observed exceedance stays within its nominal** 10%**.** Each benchmark is split four ways: score fitting, grid and budget $\beta = 0.5\,\hat{e}_T$, a reservoir the calibration draws come from, and a deployment set no selector sees, on which the selected threshold's risk is evaluated. $m = 150$, redrawn 400 times; all four selectors share the score $p_t^{(1)}$, the grid and the budget. Raw CRC has no valid expected-risk guarantee under a non-monotone loss, hence the separate row. A finite empirical-population diagnostic, not a coverage theorem; Appendix I gives the other calibration size and the local k-NN score.

			draws over β (%)				LTT
Benchmark	β	n_{dep}	plug-in	raw CRC	loss CRC	**LTT**	act (%)
MMLU-Pro	0.076	2,494	66.2	51.7	51.7	**8.5**	70
LogiQA	0.125	2,604	25.5	16.0	16.0	**1.2**	56
ARC	0.021	777	65.8	41.0	41.0	**4.0**	21
BBH	0.031	719	57.8	35.8	35.8	**7.0**	54
MuSR	0.140	228	2.8	1.5	1.5	**0.0**	22

Table 7: **The exceedance study in full**: both calibration sizes and both scores. Columns are the percentage of calibration draws whose selected threshold exceeds β on the deployment set. "harm" counts grid steps at which the *deployment* risk rises with the threshold, out of the grid size; it is small everywhere, which is why the exceedances above are attributed to finite-sample optimism rather than to non-monotonicity. A benchmark is absent at a given m when its splits cannot supply a reservoir at least $1.5m$ in size, since below that every draw is nearly the same set and the exceedance frequency degenerates. The ordering of the four selectors is the same in every row.

Benchmark	β	harm	plug-in	raw CRC	loss CRC	LTT
confidence, $m = 150$						
MMLU-Pro	0.076	0/76	66.2	51.7	51.7	8.5
LogiQA	0.125	0/84	25.5	16.0	16.0	1.2
ARC	0.021	1/63	65.8	41.0	41.0	4.0
BBH	0.031	1/34	57.8	35.8	35.8	7.0
MuSR	0.140	4/81	2.8	1.5	1.5	0.0
confidence, $m = 400$						
MMLU-Pro	0.076	0/76	64.0	54.0	54.0	10.0
LogiQA	0.125	0/84	12.8	8.8	8.8	0.8
ARC	0.021	1/63	66.5	45.5	45.5	4.0
BBH	0.031	1/34	76.2	59.0	59.0	5.5
local k-NN LCB, $m = 150$						
MMLU-Pro	0.076	0/85	62.3	49.2	49.2	8.0
LogiQA	0.125	0/98	31.8	21.5	21.5	3.5
ARC	0.021	2/82	50.2	28.7	28.7	1.2
BBH	0.031	1/40	62.5	40.2	40.2	7.5
MuSR	0.140	3/32	0.0	0.0	0.0	0.0
local k-NN LCB, $m = 400$						
MMLU-Pro	0.076	0/85	58.5	50.0	50.0	8.2
LogiQA	0.125	0/98	20.0	14.8	14.8	0.5
ARC	0.021	2/82	34.8	19.5	19.5	1.0
BBH	0.031	1/40	84.0	70.0	70.0	8.0

I.7 Exceedance on a fixed deployment set, in full

Table 7 is the study of Section 5.3 at both calibration sizes and for both scores. Two things are worth reading off it. The ordering of the four selectors is the same in every row, so the conclusion does not depend on the score, on the calibration size or on the benchmark. And the "harm" column, which counts grid steps at which the deployment risk rises with the threshold, is small throughout: the exceedances are finite-sample optimism, not non-monotonicity. That is why Section 5.3 attributes them to the former and keeps the two findings apart.

Table 8: **Adaptive stopping preserves final-round automation while reducing executed rounds** ($\beta = 0.5\,\hat{e}_T$). Each row is a policy family calibrated by the same wrapper, so all carry the same guarantee. "Rnd" is the mean number of deliberation rounds actually executed: a rule restricted to round 1 knows at round 1 whether it will defer and stops there, a rule restricted to round T always runs to T, and a first-crossing rule over all rounds runs to T before deferring. For social confidence, adaptive and round-T automation agree to within 0.6 points on all six benchmarks; for the local bound the largest gap is 1.3 points, so the equivalence is tight for the score the paper recommends and only approximate for the elaborate one. Restricting to round 1 trades automation for cost, in either direction depending on the benchmark.

Policy family	**MMLU-Pro**		**LogiQA**		**ARC**		**BBH**		**MuSR**		**GPQA**	
	Act	Rnd	Act	Rnd	Act	Rnd	Act	Rnd	Act	Rnd	Act	Rnd
Social confidence $p_t^{(1)}$	82.3	2.10	67.5	2.63	36.0	3.40	82.7	1.86	43.2	3.20	62.1	2.89
restricted to round 1	77.4	1.00	68.7	1.00	42.7	1.00	81.8	1.00	48.8	1.00	48.0	1.00
restricted to round T	82.5	4.00	67.3	4.00	36.2	4.00	82.4	4.00	42.6	4.00	62.2	4.00
Local k-NN LCB	80.9	1.82	68.7	2.26	30.8	3.55	83.8	1.77	26.5	3.62	56.9	3.22
restricted to round T	82.2	4.00	67.6	4.00	29.6	4.00	82.9	4.00	26.5	4.00	56.7	4.00

Table 9: **Automation and wrong-action rate for the local k-NN LCB score across budgets.** Left: absolute budgets. Right: budgets set relative to the final-round error, $\beta = \kappa\,\hat{e}_T$, with $\hat{e}_T$ estimated only on the splits the test does not use. Pooled over 10 splits; WA stays below β in every cell.

Dataset	$\beta = 0.05$		$\beta = 0.10$		$\beta = 0.15$		$\kappa = 0.25$		$\kappa = 0.50$		$\kappa = 0.75$	
	Act	WA	Act	WA	Act	WA	Act	WA	Act	WA	Act	WA
MMLU-Pro	70.1	4.12	86.6	8.99	95.9	14.01	64.1	3.11	80.9	6.94	91.3	11.24
LogiQA	32.2	3.78	60.5	8.75	77.4	13.77	39.9	4.89	68.7	11.00	85.6	17.01
ARC	70.2	2.61	100.0	4.32	100.0	4.32	15.8	0.41	30.8	0.91	51.6	1.74
BBH	87.0	3.54	97.9	7.45	100.0	9.10	53.0	0.98	83.8	2.75	90.2	4.37
MuSR	2.5	0.69	7.5	2.41	16.9	5.24	7.0	2.14	26.5	8.44	49.7	16.01
GPQA	6.7	1.14	34.7	6.12	62.2	11.87	15.4	2.67	56.9	10.73	75.2	15.49

I.8 What the extra rounds buy

Table 8 gives both scores at all three round restrictions; Section 5.1 quotes only the confidence rows. The local k-NN bound behaves the same way. Its adaptive and round-T automation differ by at most 1.3 points, and the adaptive rule is ahead on 4 of 6 benchmarks, so here too what first-crossing buys is not reliability, which the wrapper fixes for every row, nor automation, but executed rounds: 10–56% fewer than running to T. The round-1 restriction is reported for confidence only, because it is the score a practitioner would actually deploy at one round; on the local bound a single round leaves the neighbourhood too small for the lower confidence bound to discriminate (Appendix H).

I.9 Budget sweep

Table 9 sweeps absolute and relative budgets. Automation rises smoothly and the pooled wrong-action rate stays below β in every cell. The sweep also shows when a budget stops meaning anything: at $\beta = 0.10$ on ARC, whose $\hat{e}_T = 0.045$, every method acts on everything, because acting always already satisfies the budget. The supplement reports every policy at every budget with acted-on accuracies, executed rounds, certification counts and per-split exceedances (Appendix K.3).

Figure 5 is the graphical companion to these tables, but not a plot of them: panel (a) traces the empirical risk along the descending grid and marks the steps where it falls, which is the non-monotonicity of Section 3 on real data, and panels (b,c) give automation against the budget for each score on two benchmarks, which the sweep above tabulates for all six.

Table 13 is the only experiment here that can test Theorem 6 directly, because it is the only one where the true risk of a selected policy is known: the synthetic process admits a closed-form risk curve, so no Monte-Carlo error enters the target.

Table 10: **Other calibration routes and other policy families** ($\beta = 0.5\,\hat{e}_T$). Each cell is automation, then the number of the 10 test halves whose wrong-action estimate exceeded β. CRC attains higher automation and shows more test-half exceedances, consistent with its expected-risk rather than high-probability conditional-risk target; Bonferroni LTT shows none and pays in automation. Our adaptive LTT-calibrated extension of the singleton-set acceptance family of Wang et al. (2026a) is competitive but not better than a calibrated confidence threshold. Adaptive stability, scored by the run length of an unchanged answer, is too coarse to certify at all on five of six benchmarks.

Policy family and calibration	**MMLU-Pro**	**LogiQA**	**ARC**	**BBH**	**MuSR**	**GPQA**
Local k-NN LCB, CRC	84.5 / 5	73.5 / 4	51.6 / 2	88.8 / 5	46.0 / 3	73.6 / 5
Confidence $p^{(1)}$, CRC	85.7 / 7	71.6 / 4	57.6 / 1	88.1 / 5	56.9 / 6	73.3 / 6
Local k-NN LCB, Bonferroni LTT	76.2 / 0	62.1 / 0	14.9 / 0	61.6 / 0	12.7 / 0	16.2 / 0
Confidence $p^{(1)}$, Bonferroni LTT	77.1 / 0	60.0 / 0	3.1 / 0	60.0 / 0	24.1 / 0	31.7 / 0
Adaptive singleton-set family, LTT	82.1 / 1	66.4 / 1	26.4 / 0	66.8 / 1	39.7 / 0	61.3 / 3
Adaptive stability, LTT	0.0 / 0	0.0 / 0	0.0 / 0	0.0 / 0	0.0 / 0	7.8 / 1

Table 11: **Sensitivity to the LTT level** γ (confidence score, $\beta = 0.5\,\hat{e}_T$). A smaller γ demands more evidence per grid point and so certifies later and less often. The effect is mild where m is large and severe on ARC, whose budget is tight.

Dataset	$\gamma = 0.01$			$\gamma = 0.05$			$\gamma = 0.10$		
	Act	WA	cert	Act	WA	cert	Act	WA	cert
MMLU-Pro	78.9	6.19	10/10	80.9	6.75	10/10	82.3	7.12	10/10
LogiQA	63.4	9.89	10/10	65.9	10.48	10/10	67.5	10.90	10/10
ARC	9.4	0.26	3/10	23.5	0.74	6/10	36.0	1.15	9/10
BBH	70.2	1.69	9/10	81.6	2.44	10/10	82.7	2.63	10/10
MuSR	27.6	9.71	10/10	38.1	12.49	10/10	43.2	14.18	10/10
GPQA	45.5	5.97	10/10	56.4	8.57	10/10	62.1	10.84	10/10

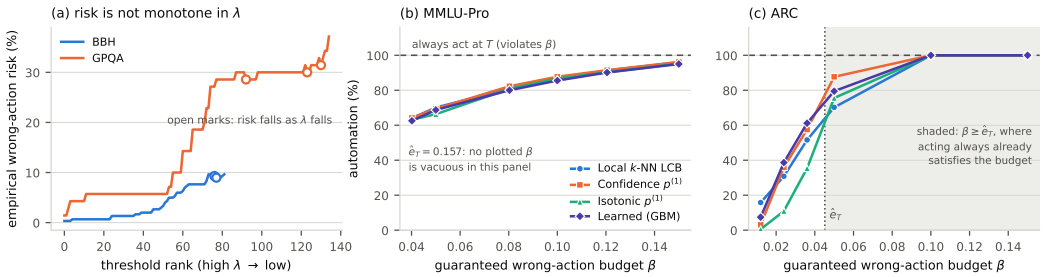

Figure 5: **(a)** Empirical wrong-action risk on $\mathcal{D}_{\text{ltt}}$ as the threshold descends. Open marks flag steps where risk falls as the threshold falls, which a monotone risk cannot do; they occur on 6 of 6 benchmarks (Table 12). **(b,c)** Automation against the budget. Each curve is separately calibrated and carries its own guarantee (Theorem 6), so higher is better; the dashed line is the automation of acting always, whose wrong-action rate is $\hat{e}_T$ and which therefore violates every budget below it. The shaded band in (c) is $\beta \geq \hat{e}_T$, where acting always already meets the budget and no score can distinguish itself; on MMLU-Pro $\hat{e}_T$ exceeds every plotted budget, so no such region appears. Scores are nearly identical on MMLU-Pro and separate on ARC, whose $\hat{e}_T = 0.045$ leaves little headroom.

Table 12: **Correctness transitions and the three terms of Theorem 1.** Left: percentage of instances whose answer flips across rounds, right-to-wrong (r→w), wrong-to-right (w→r), and right at round 1 but wrong at round T (1r,Tw). Right: the three terms of the decomposition summed over all adjacent pairs of a descending threshold grid and averaged over the 10 splits, together with the mean number of pairs per split on which the first term exceeds the other two, which is exactly where risk rises with the threshold, and the number of splits showing at least one such step, and the largest single increase in risk observed on any pair. Every benchmark shows some. All of it is computed on the calibration split $\mathcal{D}_{\mathrm{ltt}}$ of each of the 10 splits, using the same grid the method itself uses, so the risks are integer counts over $|\mathcal{D}_{\mathrm{ltt}}|$ instances; the last column gives the largest step that way, and it is worth reading before the rate beside it. On MuSR the largest step is 4.17pp because it is 4 instances out of 96, and on GPQA the entire effect on that split is a single instance out of 70. The prevalence and the size of these steps therefore carry substantial sampling uncertainty, and the 10 splits are re-splits of one dataset rather than independent replicates. What is not uncertain is their direction: were every instance's loss non-increasing, the empirical curve, being an average of non-increasing functions, could not rise at all, at any m or grid resolution. The identity of Theorem 1 holds numerically to 10^{-16} on every pair of every split.

	transitions (%)			decomposition over grid pairs, mean over the 10 splits						
Dataset	r→w	w→r	1r,Tw	ΣB	ΣF	ΣD	$B{>}F{+}D$	splits	max step (pp)	as a count
MMLU-Pro	2.4	6.9	1.6	0.014	0.049	0.158	0.4/84	4/10	0.29	3/1040
LogiQA	3.5	5.2	2.6	0.015	0.032	0.238	1.0/97	5/10	0.18	2/1087
ARC	0.6	0.5	0.6	0.005	0.002	0.040	0.7/68	6/10	0.93	3/324
BBH	1.2	1.9	0.8	0.007	0.012	0.078	0.8/43	7/10	0.66	2/301
MuSR	3.7	6.5	2.6	0.030	0.052	0.290	0.3/42	3/10	4.17	4/96
GPQA	5.1	16.7	2.7	0.041	0.151	0.274	1.0/79	7/10	1.43	1/70

Table 13: **Attained coverage on a synthetic non-monotone process.** The population risk of this process is available in closed form, so "fail" counts repetitions in which the *exact* risk of the selected policy exceeds β, with no Monte-Carlo error in the target. The grid is fixed in advance and the calibration split is redrawn independently each repetition. Coverage intervals are exact binomial (Clopper–Pearson, 95%). In every configuration the observed failure frequency is below $\gamma = 0.10$ and the coverage interval lies above the nominal $1-\gamma$. The last column shows how close the selection gets to the budget: at $\beta = 0.10$ the selected policy sits at the boundary, which is the regime in which the bound is actually tested; at $\beta = 0.05$ the grid offers nothing near the budget, so the procedure is far more conservative than it needs to be.

β	m	reps	fail	coverage	95% CI	nominal	max $R(\hat{\lambda})$
0.05	200	600	0	1.000	[0.994, 1.000]	0.90	0.001
0.05	400	600	0	1.000	[0.994, 1.000]	0.90	0.001
0.10	200	600	43	0.928	[0.905, 0.948]	0.90	0.137
0.10	400	600	32	0.947	[0.926, 0.963]	0.90	0.125

Table 14 reports the observed spread of the per-split automation difference between the local k-NN score and plain confidence. We deliberately give no interval or test: the 10 splits partition one dataset repeatedly, so the differences are dependent and resampling them as if they were i.i.d. would understate the uncertainty. The spread is very wide on ARC, where the procedure certifies on only some splits, and narrow on the two large benchmarks.

J What the observables pin down

Proposition 3 says the risk and automation curves identify the harmful mass only up to an interval. Table 15 measures how loose that is on the cached trajectories, and the answer is very loose: the interval is 2.5–23.9% wide while the harmful mass inside it is a few hundredths of a percent, a factor of 86 to 166. An operator who somehow knew both curves exactly would still not know B to within two orders of magnitude.

Table 14: **Per-split automation difference** between the local k-NN LCB and the plain confidence score at $\beta = 0.5\,\hat{e}_T$, in percentage points, over the same 10 splits. We report the observed spread only and run no test: the splits are re-splits of one dataset and are therefore not independent samples, so a bootstrap or t-interval over these differences would treat dependent quantities as i.i.d. Read the last column as the direction of the effect rather than its significance.

Dataset	mean ΔAct	min	max	local ahead on
MMLU-Pro	-1.39	-5.15	+0.51	2/10
LogiQA	+1.18	-1.27	+2.70	8/10
ARC	-5.20	-37.53	+33.51	5/10
BBH	+1.14	+0.42	+3.09	10/10
MuSR	-16.75	-38.89	+9.79	2/10
GPQA	-5.20	-24.18	+2.93	6/10

Table 15: **Exact risk and automation curves leave the harmful mass unpinned.** For every adjacent-and-wider pair of grid thresholds on every split, "B" is the true right-to-wrong mass and "interval" the width of the range Proposition 3(i) leaves once $R(\lambda)$, $R(\lambda')$ and $A(\lambda)$ are known exactly; "ratio" is their quotient. The last two columns are for the widest pairs, from the most lenient threshold towards the most cautious: "$|\Delta R|$" is the net risk change the curve shows and "churn" the gross correctness flow $B + F$ underneath it. Two readings. The observables pin B only to a range one to two orders of magnitude wider than its value, which is the empirical face of Proposition 3(i). But on these runs the churn is *small*: the risk change is carried almost entirely by deferral, D, not by correctness flips, so the aggregate curve is not hiding a large gross flow here: Corollary 5(iii) says it *can*, not that it does.

		harmful mass, %			widest pairs, %	
Dataset	pairs	B	interval	ratio	$\lvert\Delta R\rvert$	churn
MMLU-Pro	3,320	0.050	4.35	87	10.0	0.27
LogiQA	3,985	0.059	9.84	166	12.3	0.31
ARC	2,619	0.015	2.51	164	2.4	0.01
BBH	1,335	0.027	2.59	97	8.0	0.00
MuSR	1,345	0.249	23.88	96	29.4	0.42
GPQA	3,085	0.164	14.13	86	24.6	1.82

The same table reports a negative result we want on the record. Corollary 5(iii) shows the risk curve *can* be blind to an arbitrarily large correctness churn, and we expected to find some of that flow here. We do not. Even for the widest threshold pairs the churn is 0.00–1.82%, far smaller than the net risk change of the same pairs, because on these runs raising the threshold changes the risk almost entirely by *deferring* instances away the D term, rather than by overturning correctness. So the cancellation the corollary exploits is available in principle and largely unused in this data. We read this as bounding the claim rather than weakening it: the impossibility results say what the aggregate view cannot determine, and Table 15 shows the practical bite is in the width of that interval, not in hidden churn.

K Empirical non-monotonicity

Table 12 traces the decomposition of Theorem 1 on the same 10 splits and the same threshold grid the method itself uses, so it doubles as the monotonicity check: a pair on which $B > F + D$ is exactly a pair on which the risk rises with the threshold. Such pairs appear on all 6 benchmarks, on 3–7 of the 10 splits each, and the largest single increase observed is 0.0417 (on $MuSR$). The magnitudes are small, which is why conformal risk control returns similar thresholds here; the point of Theorem 1 is that the precondition is a statement about correctness transitions rather than about the score, and so cannot be checked from unlabelled deployment data alone.

K.1 What a monotone search forgoes on an exact curve

A monotonicity-licensed search cannot return an unsafe threshold: both the prefix scan and the bisection only ever return a point they have checked satisfies $R(\lambda) \leq \beta$. What non-monotonicity costs such a search is not safety but *automation*, because a disconnected feasible set has islands a prefix search never reaches. Sweeping β densely over the exact curve of each benchmark, computed on all observed trajectories with no sampling, the feasible set is disconnected on at most 3.9% of budgets, and the automation forgone relative to the best feasible threshold is positive on 1 of 6 benchmarks, reaching 0.73pp. So on these trajectories the premise fails while its utility cost is small. We report that rather than the reverse: the case for a monotonicity-free procedure here rests on validity, not on a large measured penalty.

K.2 Does modelling the transitions inside the policy help?

Theorem 1 holds for any nested pair of stopping rules, so it also describes a policy that tries to *use* the transitions rather than only to be certified against them. A first-crossing threshold asks one question, "does this state look safe". The decomposition suggests a second, "is one more round more likely to repair this answer than to break it".

We built the corresponding score. Writing $E_t = \mathbf{1}[\hat{Y}_t \neq Y]$, we fit three gradient-boosted classifiers on $\mathcal{D}_{\mathrm{mod}} \cup \mathcal{D}_{\mathrm{loc}}$ only,

$$\hat{q}_t \approx \Pr(E_t = 0 \mid u_t), \qquad \hat{p}_t^{\mathrm{fix}} \approx \Pr(E_t = 1, E_{t+1} = 0 \mid u_t),$$

$$\hat{p}_t^{\mathrm{break}} \approx \Pr(E_t = 0, E_{t+1} = 1 \mid u_t),$$

the last two being conditional versions of the F and B masses for one further round, and score a state by $S_t = \hat{q}_t - \kappa_{\mathrm{tr}}\big[\hat{p}_t^{\mathrm{fix}} - \hat{p}_t^{\mathrm{break}} - \eta\big]^+$: act when the current answer looks right *and* another round is not expected to help. Thresholding S_t gives an ordered family, so Corollary 11 certifies it with no new theory and the comparison in Table 16 is at a matched guarantee.

The conditioning variable u_t is a deliberately compact summary of the transcript, not the transcript: it is the round's six features of Appendix G, their one-step change, and the round index. Everything below is therefore a statement about this summary and this model class. A policy with access to the raw agent messages might do better; we did not test one.

Three win criteria, and they disagree. Table 16 reports automation and mean executed rounds, and we count wins against *both* scalar scores at once under each of three criteria. The transition-aware score achieves higher automation in 8 of 18 benchmark-budget cells, higher automation per executed round in 7, and a Pareto improvement in both quantities in 6. The second and third counts depend on how a split that certifies nothing is charged, and we charge it the same horizon a deferring policy is charged on a certified split, as elsewhere in the paper; scoring such a split at zero rounds instead would flatter whichever score certifies least often. The criteria disagree at the margin: on LogiQA at $\kappa = 0.75$ it automates more than either scalar score but the local bound reaches almost the same automation in fewer rounds, so it wins on automation and loses on efficiency, and on MuSR at $\kappa = 0.50$ it automates more while executing more rounds than confidence, so it is not a Pareto improvement. Reporting a single "wins" count would hide that. Under every one of the three criteria the score wins fewer than half the cells, so we claim no consistent overall advantage.

Variants. We also fit a hard gate, which sets the score to $-\infty$ whenever $\hat{p}_t^{\mathrm{fix}} - \hat{p}_t^{\mathrm{break}} > 0$ and so refuses to act on such states rather than merely discounting them, the refusal being absolute because the grid is built from finite scores only and no threshold on it can select a gated state; a positive cost term $\eta = 0.02$, charging one round of deliberation against the expected repair; and a horizon of all later rounds, replacing "repaired at $t+1$" with "correct at some later round" and "broken at $t+1$" with "wrong at every later round". All three appear in Table 16 and none changes the verdict, so we keep the calibrated scalar threshold as the method of the paper and report this as a negative result.

What the reference row is, and is not. The last row of Table 16 stops at the first round whose answer is correct and escalates when no round is, so it never acts on an answer it would get wrong. It is a zero-error perfect-information reference. It is *not* an upper bound on automation under a positive

wrong-action budget: a calibrated policy may spend that budget acting on instances it gets wrong, and at the loosest budget several do so often enough to automate more than the reference does, in 8 of the cells. Where the comparison is informative is at $\kappa = 0.50$, where the reference automates 4.1 to 60.2 points more than the better of the two scalar scores while making no wrong actions at all. Headroom exists there, and the models above do not reach it.

Two things make this informative rather than merely negative. First, the guarantee is unaffected: the wrapper certifies the learned policy exactly as it certifies a threshold, which is what Theorem 1 being a statement about nested rules buys. Second, the failure is localised. It is not that stopping-time structure is worthless, but that this compact history summary and this model class did not capture the headroom the reference exposes. What a richer representation of the deliberation would do is open.

Table 16: **Modelling the transitions of Theorem 1 inside the policy does not reliably beat a calibrated scalar.** Each cell is automation, then mean executed rounds, pooled over the same 10 splits, and then in brackets the number of those splits on which LTT certified any threshold, shown only when it is below 10: so ARC at $\kappa = 0.25$ reading "3.1/3.96 [1/10]" is 3.1% automation at 3.96 mean rounds, with 1 of 10 splits certified. A split certifying nothing defers everything and is charged the horizon a deferring policy is charged elsewhere, not zero. Every row but the last is calibrated by the same wrapper and carries the same guarantee, so automation is comparable across them. The transition-aware score is $S_t = \hat{q}_t - \kappa_{\mathrm{tr}}[\hat{p}_t^{\mathrm{fix}} - \hat{p}_t^{\mathrm{break}} - \eta]^+$, where $\hat{q}_t$ estimates $\Pr(\text{right at } t \mid u_t)$ and the two transition models estimate the F and B masses of Theorem 1 for one further round, all three being gradient-boosted classifiers fitted on $\mathcal{D}_{\mathrm{mod}} \cup \mathcal{D}_{\mathrm{loc}}$ only from a compact summary u_t (the round's six transcript features, their one-step change, and the round index). Against *both* scalar scores at once it achieves higher automation in 8 of 18 benchmark-budget cells, higher automation per executed round in 7, and a Pareto improvement in automation and rounds together in 6; the three criteria disagree because its gains in automation are usually bought with extra rounds. The last row stops at the first correct round, so it never acts on an answer it would get wrong. It is a zero-error perfect-information reference and *not* an upper bound on automation: a calibrated policy is permitted to spend the budget on instances it gets wrong, and does so often enough to exceed the reference in 8 cells, all at the loosest budget. At $\kappa = 0.50$ the reference automates 4.1 to 60.2 points more than the better scalar score, so headroom exists there; these transition models do not reach it.

Policy	MMLU-Pro	LogiQA	ARC	BBH	MuSR	GPQA
$\kappa = 0.25$						
Confidence $p_t^{(1)}$	64.2/2.65	39.9/3.25	3.1/3.96 [1/10]	51.9/2.78 [7/10]	17.4/3.72	31.7/3.56
Local k-NN LCB	64.1/2.52	39.9/3.27	15.8/3.77 [8/10]	53.0/2.77 [7/10]	7.0/3.90 [8/10]	15.4/3.86 [8/10]
Transition-aware S_t	65.0/2.44	43.3/2.98	11.0/3.79 [9/10]	65.3/2.37	16.9/3.76	23.8/3.69
hard gate	63.5/3.79	41.7/3.93	10.0/3.84	62.0/3.19	18.8/3.80	28.2/3.77
cost term $\eta = 0.02$	64.8/2.44	43.4/2.97	8.6/3.83 [9/10]	65.3/2.37	16.9/3.76	23.9/3.69
all later rounds	64.8/2.45	43.3/2.98	11.4/3.78 [9/10]	65.4/2.37	16.9/3.76	23.7/3.70
Zero-error reference	86.3/1.49	79.0/1.69	96.3/1.12	92.7/1.24	69.3/2.01	76.1/1.93
$\kappa = 0.50$						
Confidence $p_t^{(1)}$	82.3/2.10	67.5/2.63	36.0/3.40 [9/10]	82.7/1.86	43.2/3.20	62.1/2.89
Local k-NN LCB	80.9/1.82	68.7/2.26	30.8/3.55 [9/10]	83.8/1.77	26.5/3.62	56.9/3.22
Transition-aware S_t	80.8/1.87	70.0/2.20	37.2/3.22	79.1/1.95	44.3/3.25	52.6/3.12
hard gate	80.0/3.68	67.9/3.83	35.3/3.49	78.9/2.91	43.5/3.46	48.2/3.58
cost term $\eta = 0.02$	80.8/1.87	70.0/2.20	37.2/3.23	79.1/1.95	44.1/3.25	52.2/3.13
all later rounds	80.9/1.87	70.0/2.20	37.2/3.22	79.1/1.95	44.0/3.25	51.6/3.15
Zero-error reference	86.3/1.49	79.0/1.69	96.3/1.12	92.7/1.24	69.3/2.01	76.1/1.93
$\kappa = 0.75$						
Confidence $p_t^{(1)}$	91.5/1.76	85.0/2.02	57.6/2.66 [9/10]	89.1/1.63	69.9/2.44	76.2/2.55
Local k-NN LCB	91.3/1.48	85.6/1.71	51.6/2.96 [9/10]	90.2/1.55	49.7/3.43	75.2/2.71
Transition-aware S_t	89.9/1.57	85.8/1.75	59.6/2.60	89.8/1.56	68.7/2.72	64.9/2.77
hard gate	90.3/3.59	85.6/3.68	55.5/3.17	88.7/2.65	69.6/3.05	65.0/3.35
cost term $\eta = 0.02$	89.9/1.57	85.7/1.75	59.6/2.60	89.8/1.56	68.9/2.71	64.9/2.77
all later rounds	89.9/1.58	85.6/1.76	59.6/2.61	89.8/1.57	68.9/2.71	65.6/2.75
Zero-error reference	86.3/1.49	79.0/1.69	96.3/1.12	92.7/1.24	69.3/2.01	76.1/1.93

K.3 Full results, and where they are

Every policy on every benchmark at every budget, with acted-on accuracy, mean executed rounds, certification counts and per-split exceedance, runs to six pages of numbers that answer questions nobody asks while reading. They are therefore in the supplement rather than here: the rendered tables are `FULL_TABLES.pdf` at the top of the archive, the LaTeX sources are `tables/tab_full.tex`, and `code/gen_tables.py` regenerates both from `results/canonical_iclr.json`. Nothing in them is used above without being quoted; the summary a reader needs is Table 3 and Figure 4, and `README.md` maps each main-text figure to the table that carries its exact numbers.

L Horizon scaling

Section 5.1 reads seven horizons off two runs. This section states exactly how, defines the four quantities the figure and Table 17 report, and describes the code self-revision setting, which is the only place in the paper where correctness is decided by running a program rather than by comparing a letter.

Table 17: **Every horizon, both settings.** Each row truncates the same $T = 8$ run to its first T rounds, so instances, model and sampling are identical down the column and only the horizon changes. "acc_1" and "acc_T" are accuracy at round 1 and at the horizon; "gain" is their difference; "r→w" is the share of instances whose answer is overturned from correct to wrong at some adjacent round pair inside the horizon, not the share still wrong at T since some are repaired, and "w→r" the converse; "non-mono." is the share of size-150 calibration draws that are *non-monotone*, meaning their empirical risk curve rises somewhere on the grid, with an exact Clopper–Pearson interval over the 1,000 draws; and H_T is the largest upward step of the full-set curve, in percentage points. The runs are `solo_MMLUPro_T8_dist.json`, `coderepair_T8.json`, in that order. The gain saturates while r→w does not. In code self-revision every draw is monotone at every horizon and H_T is identically zero, so nothing in the last two columns reveals the growth in r→w. A r→w transition is necessary but not sufficient for a threshold-wise loss reversal, so the r→w column measures answer instability rather than counting premise violations.

T	acc_1	acc_T	gain	r→w	w→r	non-mono. (%)	95% CI	H_T
MMLU-Pro, self-revision ($n = 1000$)								
2	78.6	79.1	+0.5	1.9	2.4	33	[30, 36]	0.00
3	78.6	79.8	+1.2	3.4	4.6	49	[46, 52]	0.00
4	78.6	80.1	+1.5	4.5	6.3	68	[65, 71]	0.10
5	78.6	80.5	+1.9	5.7	7.4	69	[66, 72]	0.00
6	78.6	80.9	+2.3	6.8	8.9	75	[72, 77]	0.00
7	78.6	80.9	+2.3	7.8	9.7	82	[79, 84]	0.10
8	78.6	80.8	+2.2	8.7	10.8	90	[88, 92]	0.00
Iterative code self-revision ($n = 591$)								
2	77.5	78.2	+0.7	0.2	0.8	0	[0, 0]	0.00
3	77.5	78.0	+0.5	0.7	1.2	0	[0, 0]	0.00
4	77.5	78.5	+1.0	0.8	1.9	0	[0, 0]	0.00
5	77.5	78.5	+1.0	1.0	2.2	0	[0, 0]	0.00
6	77.5	78.0	+0.5	1.4	2.2	0	[0, 0]	0.00
7	77.5	78.3	+0.8	1.5	2.4	0	[0, 0]	0.00
8	77.5	78.0	+0.5	2.0	2.5	0	[0, 0]	0.00

L.1 Horizon scaling protocol

Prefixes, not separate runs. Each run generates $T = 8$ rounds once per instance. The horizon-T analysis is the same transcript truncated to its first T rounds: identical instances, model, prompts and sampled tokens. Nothing is regenerated between horizons, so a difference between two rows cannot come from a different sample of instances or from decoding variation, only from the extra rounds themselves. The price of this design is that the rows are not independent, so we read the trend and do not attach a test to it.

The four quantities. Fix a horizon T and write $c_t(i) \in \{0, 1\}$ for whether instance i's round-t answer is correct. *Accuracy gained* is $\frac{1}{n}\sum_i c_T(i) - \frac{1}{n}\sum_i c_1(i)$, the net change a caller who always waits for round T would see. *Correct answers broken* counts an instance if the answer is ever overturned from correct to wrong inside the horizon, $\frac{1}{n}\sum_i \mathbf{1}\big[\exists\, t < T :\ c_t(i) = 1,\ c_{t+1}(i) = 0\big]$, and the reverse direction is reported beside it. We count any adjacent-round reversal rather than only $c_1 = 1$, $c_T = 0$ because a stopping rule can act at any round, so an answer broken at round 3 and repaired by round 8 is still a wrong action available to a caller who stops at 3; by the same token the count is not a count of answers lost by round T, and we do not call it one. Such transitions are the trajectory-level events that can contribute to the break mass B of Theorem 1, for stopping-rule pairs that act on opposite sides of one; they are not themselves B, as Appendix L.2 spells out. Reporting both directions keeps the net gain from being read as if nothing were lost. Both are shares of instances and share the axis of Figure 2's first two panels.

A *non-monotone draw* is defined at the level of a calibration set, not an instance. Draw $m = 150$ instances without replacement, take 100 equally spaced quantiles of that draw's own confidence scores as the threshold grid, form the empirical risk $\widehat{R}$ on it, and call the draw non-monotone if $\widehat{R}$ rises anywhere as the threshold is raised, that is if $\widehat{R}(\lambda_{j-1}) - \widehat{R}(\lambda_j) > 0$ for some j with the grid enumerated from the most conservative candidate down. The reported share is over 1000 independent draws at a fixed seed. This is the event an operator would actually witness on their own calibration set, which is why it is a share of draws rather than of instances. Finally H_T is the largest such rise computed once on the full set instead of on a draw, in percentage points: the aggregate diagnostic a practitioner would consult, and by Corollary 5 the one that can sit at zero while the trajectory quantity grows.

The code self-revision setting. Instances are the union of HumanEval and the MBPP test split, $n = 591$ after dropping problems whose reference tests do not execute in our sandbox. The model receives the problem statement and, for MBPP, the required function signature, and returns a complete function plus a verbalized confidence in $[0, 1]$. Correctness is decided by executing the benchmark's own tests against the returned function in a subprocess with a fixed preamble of standard imports and a timeout; a crash, a timeout and a failed assertion are all incorrect. Each later round shows the model its own previous program together with the fact that it may be wrong, so revision is self-driven exactly as in the multiple-choice settings. The benchmark tests are used only for post-hoc correctness evaluation and are never shown to the model or fed back as a signal during revision; the signature is passed because without it the model writes a correct function under another name or argument order, which measures specification ambiguity rather than revision ability. We therefore call the setting self-revision rather than repair: "repair" normally implies that a failing test or an error message is fed back, and here nothing is.

What the table shows. On MMLU-Pro the gain saturates by $T \approx 6$ while broken answers keep climbing to 8.7%, and the share of affected draws reaches 90%, yet H_T never exceeds 0.10pp. Code self-revision is the sharper case: broken answers grow from 0.2% to 2.0% while the gain stays between $+0.5$ and $+1.0$ points, and both risk-curve diagnostics are identically zero at every horizon.

L.2 Interpreting broken answers

A right-to-wrong transition is not the same event as a threshold-wise loss reversal, and we do not want the first read as the second. The loss at threshold λ is $\mathbf{1}[\text{act} \wedge \text{wrong}]$, so a reversal of the *loss* needs two thresholds that act at the two different rounds. A threshold acts at a round only if that round's score exceeds every earlier score in the trajectory, so when no threshold separates the two rounds of a transition (the later score does not clear the earlier one, or both rounds fall on the same side of every grid point), the loss is flat there at every threshold. The transition is therefore necessary for a pointwise violation but not sufficient. What the last two columns license is the statement we make, that answer instability can grow while aggregate risk diagnostics stay nearly flat, and not the stronger claim that each broken answer breaks CRC's premise.

Table 18: **Reordering the fixed sequence does not help** ($\gamma = 0.10$, pooled over 10 splits). Every row splits the benchmark four ways: grid, a pilot split that fixes the testing order, the calibration labels, and a deployment set none of the three sees. "thr" is the paper's order, the grid from the most conservative candidate down; "util" orders the candidates the pilot calls feasible by decreasing pilot automation; "risk" orders all candidates by increasing pilot risk. "over" is the share of splits whose deployed policy exceeds β and "cert" the mean length of the certified prefix. Utility ordering loses automation in every cell because the sequence ends at its first non-rejection; risk ordering reproduces the threshold order up to noise and never beats it.

			threshold order			utility order		risk order	
Dataset	κ	β	Act	over	cert	Act	Δ	Act	Δ
MMLU-Pro	0.25	0.040	66.9	20	29.0	20.4	-46.5	66.9	+0.0
MMLU-Pro	0.50	0.080	83.5	30	45.4	25.0	-58.4	83.4	-0.1
MMLU-Pro	0.75	0.120	92.3	10	56.6	46.3	-46.0	92.3	+0.0
LogiQA	0.25	0.061	39.4	0	21.7	8.3	-31.1	39.2	-0.2
LogiQA	0.50	0.123	68.2	0	40.6	20.6	-47.6	68.2	+0.0
LogiQA	0.75	0.184	85.9	10	55.6	34.3	-51.6	85.9	+0.0
ARC	0.25	0.012	9.3	10	1.1	3.0	-6.3	8.2	-1.2
ARC	0.50	0.024	41.8	10	8.3	13.7	-28.1	40.5	-1.3
ARC	0.75	0.036	71.1	10	27.0	42.9	-28.2	68.9	-2.1
BBH	0.25	0.020	66.6	20	3.7	7.8	-58.8	66.2	-0.3
BBH	0.50	0.041	84.6	10	13.4	9.0	-75.6	83.6	-1.0
BBH	0.75	0.061	91.0	10	21.2	18.6	-72.4	90.7	-0.3
MuSR	0.25	0.090	18.8	10	11.8	3.7	-15.0	18.4	-0.4
MuSR	0.50	0.181	42.8	10	25.1	13.6	-29.2	41.4	-1.4
MuSR	0.75	0.271	72.0	10	44.6	20.4	-51.7	70.3	-1.8
GPQA	0.25	0.072	39.8	0	14.1	8.0	-31.9	36.3	-3.5
GPQA	0.50	0.143	63.3	10	32.3	13.0	-50.3	61.7	-1.6
GPQA	0.75	0.215	80.9	0	45.9	38.8	-42.2	79.8	-1.1

M REORDERING THE FIXED SEQUENCE

Fixed-sequence testing controls the error over the whole rejected prefix, and it does so for any order fixed before the calibration labels are read. The prefix property matters here: a caller may inspect the certified prefix afterwards and take its highest-automation member, with no multiplicity correction, because if any member of the prefix were unsafe then the first unsafe candidate in the order was falsely rejected, an event of probability at most γ. So the order is a free design choice, and Theorem 1 says the safe candidates need not form an interval of the threshold grid. That invites a data-driven order.

Table 18 reports two. Each benchmark is split four ways (grid, a pilot split that fixes the order, the calibration labels, and a deployment set none of the three sees), and the order is a function of the pilot split alone, so validity is untouched in every row.

Ordering by pilot automation is much worse everywhere, by 6 to 76 points. The mechanism is structural rather than statistical: the sequence terminates at its first non-rejection, so putting the most aggressive candidate first ends it almost immediately, and the certified prefix collapses. Pilot optimism, which Section 5.3 measures directly, makes the leading candidate unsafe often enough for this to dominate.

Ordering by pilot risk never wins either, at best $+0.0pp$. Pilot risk is nearly monotone in the threshold, so this order mostly reproduces the threshold order, and where it does not it is following estimation noise. Corollary 5 is the reason not to expect better: the aggregate risk information a pilot split provides does not identify the trajectory structure that would justify a different order. We report this as a negative result: among the orders we tested, the threshold order is the best, and the theory says why aggregate risk information gives a reordering little to work with. It does not show that no risk-based ordering could ever help.

Table 19: **With instances, elicitation and generation budget held fixed and only the agent configuration varied, every protocol exhibits right-to-wrong transitions.** All rows use the 707 MMLU-Pro questions complete in *every* run; "raw" and "compl." give each run's size before it. All runs elicit a verbalized distribution with `maxTokens` $= 4096$ and $T = 4$; Δ carries a bootstrap interval. "subsamp." is the share of size-m draws with a harmful threshold step at $m = 150/300/600$, "pooled" whether the full-set curve has one. Completion is instance-dependent, so this ranks nothing.

	run size		accuracy (%)			transitions (%)		non-monotone	
Protocol	raw	compl.	r1	r4	Δ [95%]	r→w	w→r	subsamp.	pooled
Single agent, self-revision	1000	999	86.6	88.3	+1.7 [+0.0,+3.4]	3.5	5.1	58/53/28	yes
Three agents, ours	1000	759	89.3	90.4	+1.1 [-0.4,+2.7]	1.8	3.0	53/27/12	no
Three agents, disjoint	1000	896	86.0	85.0	-1.0 [-2.8,+0.8]	5.2	4.7	92/95/80	yes

N PROTOCOLS, AND WHAT A ROUND COSTS

Table 20: **A single agent revising its own answer also shows right-to-wrong transitions, so peer disagreement is not necessary, and a round has a price.** Each row is one deliberation protocol run end to end. "→" columns count the instances whose correctness changes at least once across the four rounds, in the stated direction; these are the transition events from which the masses B and F of Theorem 1 are constructed, so a protocol with none of them cannot exhibit non-monotone risk. *The rows here are not a controlled comparison*: each is its own complete-case set and they differ in elicitation, instance count and model family as well as in agent count. Table 19 is the comparison, on one common instance set; this table reports each run as run. The last two columns price a round: mean output tokens per instance-round, and cost in units of 10^{-3} USD per instance-round at on-demand rates, so a rule that executes one round fewer saves that much. Rows marked "votes" record only each agent's chosen option rather than a verbalized distribution, which makes the top-probability score take at most $N + 1$ values and, at $N = 1$, constant; they are reported for their transition rates and costs, not as calibration benchmarks. Instance counts are after dropping instances where an agent returned no parsable answer in some round; "drop" is that share of the raw run, which is substantial for the two triples because one member of each frequently exhausted the token budget before emitting a parsable answer.

			accuracy			transitions (%)		per round	
Protocol	n	drop	r1	r4	Δpp	r→w	w→r	tok_{out}	mUSD
Single agent, MMLU-Pro (dist)	999	0%	0.781	0.813	+3.2	4.8	7.9	585	3.76
Original triple, MMLU-Pro (dist)	759	24%	0.893	0.905	+1.2	1.7	2.9	3370	19.31
Disjoint triple, MMLU-Pro (dist)	896	10%	0.809	0.794	-1.6	7.1	5.9	1795	3.67
Single agent, BBH (dist)	2,219	0%	0.911	0.918	+0.7	3.7	4.6	480	3.08
Original triple (vote), MMLU-Pro	460	23%	0.861	0.883	+2.2	2.6	4.6	3296	17.03
Single agent (vote), MMLU-Pro	2,999	0%	0.794	0.791	-0.3	6.2	5.9	512	3.26
Disjoint triple (vote), BBH	2,152	3%	0.906	0.911	+0.5	3.1	3.6	1469	2.86
Single agent (vote), BBH	2,219	0%	0.908	0.908	-0.0	3.7	3.8	405	2.59

Table 19 is the matched comparison Section 5.1 quotes: the 707 questions complete in all three distribution runs, with elicitation, horizon and generation budget fixed and only the agent set varying. Table 20 reports every deliberation protocol we ran end to end. Three points are worth separating.

N.1 PAIRWISE COMPARISONS AND PAIRED UNCERTAINTY

Table 21 restricts each pair of protocols to the questions complete in both, a larger set than the three-way intersection, and resamples instances jointly so the differences are paired. Sample size is what decides here. On the three-way set of 707 questions most paired differences have intervals containing zero, so we read that table's ordering as descriptive. On the two-arm sets, which run to 895 questions, the unadjusted paired intervals for the transition-rate differences exclude zero and agree in direction with the levels: the single agent's right-to-wrong rate is above the original triple's and below the disjoint triple's, and the disjoint triple's rate is above the original triple's. This is not

Table 21: **Pairwise complete-case comparisons, with paired bootstrap intervals**. Each row restricts two protocols to the questions complete in *both*, a larger set than the three-way intersection of Table 19, and resamples instances jointly so the difference is paired. On these larger sets the unadjusted paired intervals exclude zero on their respective sets and agree in direction with the levels; this is not a global ranking, since each row conditions on a different complete-case population and the intervals are not adjusted for multiplicity. The levels themselves move with the restriction set, which is what instance-dependent completion produces, so numbers are comparable within a row and not across rows.

Pair	n	r→w (%)	Δr→w [95%]	Δacc (%)	difference [95%]
1 agent vs. 3 ours	759	3.6 / 1.7	+1.8 [+0.4,+3.4]	+2.0 / +1.2	+0.8 [-1.3,+2.8]
1 agent vs. 3 disjoint	895	4.6 / 7.2	-2.6 [-4.6,-0.4]	+2.9 / -1.6	+4.5 [+1.9,+6.9]
3 ours vs. 3 disjoint	707	1.8 / 5.2	-3.4 [-5.1,-1.7]	+1.1 / -1.0	+2.1 [-0.1,+4.4]

a global ranking. Each row conditions on a different complete-case population, the intervals are not adjusted for the three comparisons, and the levels shift with the restriction set, which is what instance-dependent completion produces; so numbers are comparable within a row and not across rows, and none of this isolates agent count from model quality.

All reported bootstrap intervals, here and in Table 19, are unadjusted percentile intervals from $2{,}000$ instance-level resamples with seed 0; paired comparisons resample the aligned instances jointly, so the resample index is shared across the two arms and the difference inherits the pairing.

Where the non-monotonicity lives. The crossover quoted in Section 5.3 is measured here. For each protocol run we draw calibration sets of several sizes, build the grid on each draw and count the draws whose empirical risk curve rises somewhere with the threshold. Across the runs with a graded score the share of draws whose curve contains a harmful step is 38–98% at $m = 150$, and the curve over each run's full set contains none on 3 of 4. The share does not move with m in one direction: on MMLU-Pro it falls as m grows, on BBH it rises, so we report the level at one m and note the heterogeneity rather than claiming a trend. The vote runs show almost nothing at any size, and the reason is mechanical rather than substantive: with N agents and a one-hot record the top-probability score takes at most $N + 1$ values, so its risk curve has too few distinct thresholds for a local reversal to appear. The phenomenon needs a graded confidence signal, which is the elicitation the main protocol uses.

Transitions, not agents. The two columns that matter for Theorem 1 are the transition masses, and the single-agent rows carry them: an iterative predictor with no peers at all still talks itself out of correct answers. That supports one direction, that peer disagreement is not *necessary* for non-monotone stopping risk. It does not support the converse. The rows differ in elicitation, in instance count and in which model family is involved as well as in agent count, so the ordering of the right-to-wrong column across rows is not a controlled comparison and we do not read it as one: Table 19 is that comparison, holding instances, elicitation and generation budget fixed and varying only the agent set.

Elicitation matters more than we expected, and it cuts against us. The rows marked "votes" record only each agent's chosen option, so the social belief is a vote fraction and the top-probability score takes at most $N + 1$ distinct values, and exactly one when $N = 1$. Under that elicitation the plain confidence score, which wins most cells in the paper's main comparison, frequently certifies nothing at all, while the learned predictor, which reads all six features and the round index, still certifies and automates. This is a genuine qualification of Section 5.3: "simple beats elaborate" holds when the protocol supplies a graded confidence signal, and fails when it supplies only votes. The wrapper behaves correctly in both cases, refusing to certify rather than acting unsafely, which is the intended failure mode, but a practitioner choosing a score should read the two regimes separately. Three of the rows are distribution runs on the same question list with the same elicitation and the same generation budget. Each row here reports its own complete cases, so the counts differ: after dropping instances where an agent returned no parsable answer the single-agent run retains 999 and the original triple 759. The comparison between them is Table 19, which restricts all three to the 707

questions complete in every run, and Table 21, which restricts each pair to the questions complete in both.

Rounds are not the unit of cost. The auxiliary runs record per-call tokens, so a round can be priced. Round 1 accounts for only 18% of the four-round token bill, because every later round carries all of the previous round's responses in its prompt, so rounds are not interchangeable units and a round count understates what stopping early saves. Quantifying that for the main experiment requires a token profile it does not have: the cached transcripts record no token counts. Applying the profile measured on the original triple here as a *proxy*, and pricing each policy by its executed-round distribution rather than its mean, gives 16 to 57 percent fewer tokens against 15 to 54 percent fewer rounds. We report that as an indication of direction and magnitude, not as a measurement of the main experiment, and we do not put it in the abstract; a per-benchmark figure would need each benchmark re-run with token accounting.

The price of a round, and the limit of that estimate. Round 1 carries no prior-round context, so it is much the cheapest: the cumulative share of the four-round token bill through rounds 1 to 4 is roughly 18, 46, 72 and 100 percent. Because a calibrated policy stops at different rounds on different instances we price it by its executed-round distribution rather than by its mean round count. For the rows in this table the profile and the policy come from the same run. For the main experiment they cannot: the cached transcripts record no token counts, so Section 5.1 applies one auxiliary profile to all six benchmarks. That is a *proxy*. It is a defensible one, since the profile is dominated by prompt growth that the protocol fixes rather than by anything task-specific, but it is not a measurement, and a per-benchmark number would require re-running each benchmark with token accounting. Latency behaves the other way: output tokens per round fall slightly with t, so wall-clock per round is roughly flat and a round saving translates to latency almost one for one. Costs are on-demand Bedrock rates at the time of the runs and should be read as ratios rather than absolute prices.

Two caveats on these runs, which the main-text claims are worded to respect. First, they are auxiliary: they match the cached protocol's generation budget ($\texttt{maxTokens} = 4096$) but record votes rather than distributions except where noted, and they cover fewer instances, so they are evidence about the phenomenon and about cost rather than a second copy of the main experiment. Second, instances where an agent returned no parsable answer in some round are dropped, since a social belief formed from a different number of agents is not comparable. The dropped share reaches about a quarter on the runs containing DeepSeek-R1, which routinely spent its whole budget reasoning without emitting a parsable answer, and is near zero elsewhere; the table gives both the retained count and the share. Missingness may be instance-dependent, so cross-protocol comparisons should use only the common complete-case set and be interpreted accordingly, which is a further reason not to read the ordering across rows as a controlled comparison.

O SUBGROUP SAFETY

Table 22: **Subgroup risk under a pooled threshold, MMLU-Pro** ($\kappa = 0.5$, pooled $\beta = 0.080$). One threshold is calibrated on pooled data, then evaluated per subgroup. Bold marks subgroups whose own wrong-action rate exceeds the pooled budget. Pooled validity does not imply subgroup validity.

Subgroup	n_g	Act	Acc|Act	WA	WA$/\beta$
chemistry	1132	92.0	93.0	6.47	0.81
economics	844	86.5	91.2	7.61	0.95
engineering	969	77.8	87.9	**9.42**	**1.17**
health	818	73.9	88.2	**8.73**	**1.09**
law	1101	46.1	78.1	**10.08**	**1.25**
math	1351	95.1	96.5	3.30	0.41
physics	1299	90.9	94.2	5.28	0.66
psychology	798	76.7	91.2	6.78	0.84

Tables 22 and 23 calibrate a single threshold on pooled data and then evaluate each subgroup separately. Pooled validity does not transfer. On MMLU-Pro three of eight domains exceed the

Table 23: **Subgroup risk under a pooled threshold, BBH** ($\kappa = 0.5$, pooled $\beta = 0.041$). One threshold is calibrated on pooled data, then evaluated per subgroup. Bold marks subgroups whose own wrong-action rate exceeds the pooled budget. Pooled validity does not imply subgroup validity.

Subgroup	n_g	Act	Acc|Act	WA	WA/β
date_understanding	250	93.9	93.6	**5.97**	**1.47**
geometric_shapes	250	87.8	86.5	**11.87**	**2.93**
logical_deduction_five_objects	250	100.0	100.0	0.00	0.00
logical_deduction_seven_objects	250	99.6	100.0	0.00	0.00
movie_recommendation	249	13.2	92.6	0.97	0.24
penguins_in_a_table	146	99.7	100.0	0.00	0.00
reasoning_about_colored_objects	250	99.4	99.8	0.16	0.04
salient_translation_error_detection	250	50.5	85.5	**7.33**	**1.81**
tracking_shuffled_objects_five_objects	250	99.9	100.0	0.00	0.00
tracking_shuffled_objects_seven_objects	250	100.0	100.0	0.00	0.00

pooled budget, with `law` reaching 10.1% against $\beta = 0.080$; on BBH the effect is larger, with `geometric_shapes` at 11.9% against $\beta = 0.041$, or 2.9 times the budget, at acted-on accuracy 86.5% where the pooled figure is 96.7%. The mechanism is that the two-dimensional state ranks instances by a quantity whose relation to correctness differs across subtasks, so a threshold that is safe on average is unsafe where the ensemble is confidently wrong. Remark 2 gives the remedy, calibrating per group at level γ/J, and it requires no additional smoothness or representation assumptions, but it needs enough calibration data per group, which BBH's 146–250 instances per subtask do not comfortably provide. We report this as a limitation rather than solving it.