RETRYPREC: PRECISION AS A RETRY-TIME RESOURCE UNDER AN EXTERNAL VERIFIER

Anonymous authors
Paper under double-blind review

ABSTRACT

A quantized language model is often served next to its BF16 original, and a task with a verifier lets a failed attempt be tried again. After a low-precision failure, should the next attempt stay in low precision or switch? The usual rule ranks the two copies by average accuracy and repeats the winner. However, a failure is a selection event: the problems it leaves open lean toward those on which the quantized copy is weak, and at our reference setting a BF16 attempt solves more problems per second from the second failure on. We show that exchanging one low-precision attempt for a high-precision one changes the solve rate by the mean gap times the fraction of problems still open, plus a covariance between staying unsolved and the precision gap. We propose RetryPrec, which runs one block of low-precision attempts and then one block of high-precision attempts, with the block lengths chosen on held-out problems by an unbiased estimate of the solve rate. One such schedule is optimal among the policies that react only to failures, and we derive how failures move the ratio of the two hazards. On twelve settings at matched wall clock RetryPrec reaches 78.78 points, against 75.95 for the fixed low-precision schedule and 77.60 for the best of sixteen other strategies. A test registered before the experiments shows that mixing precisions adds little once each problem uses its better one, so most of the attainable gain lies in choosing a precision for each problem, which failures partly reveal. When the budget also charges the memory held by both checkpoints, the schedule stays in low precision.

1 INTRODUCTION

Large language models are often served in low precision. Post-training quantization such as GPTQ and AWQ cuts memory and latency while keeping most of the accuracy of the BF16 weights (Frantar et al., 2023; Lin et al., 2024; Kurtic et al., 2025). On tasks with a verifier, such as code with unit tests or math with a checkable answer, a failed attempt can be tried again (Chen et al., 2021; Brown et al., 2024). A server that keeps both copies, or serves several bit widths from one set of weights (Park et al., 2024; Nair et al., 2025), then faces a small but frequent decision: after the quantized copy fails, should the next attempt use it again, or the BF16 weights? The common practice keeps using the copy with the better average accuracy (Jin et al., 2024; Kurtic et al., 2025). This rule fits the first attempt, which faces every problem. A retry faces only the problems that the first attempt left open.

Those problems are not a random sample. A problem that survives a failed low-precision attempt is more likely to be one on which the quantized copy is weak, so each failure moves the remaining problems away from what the average describes. Survival analysis calls this a selection effect: in a population of mixed risks the event rate among the survivors falls over time, because the units with the highest rates leave first (Vaupel et al., 1979). We measured it with 64 attempts per problem and configuration. At our reference setting, Qwen2.5-32B with GPTQ-W4 on code, the yield per second of one more quantized attempt, the chance that it solves an open problem divided by its cost, falls from 0.1176 to 0.0295 after two failures, while that of a switch to the BF16 copy falls only from 0.0765 to 0.0328 and overtakes it (Fig. 1b). The better retry therefore depends on how the two success probabilities vary together across problems, which the two means cannot show on their own.

Under two assumptions, independent restarts and a verifier that accepts only correct answers, this dependence has an exact form. A schedule with m low-precision and n high-precision attempts solves a fraction $1 - \mathbb{E}[(1 - q_L)^m (1 - q_H)^n]$ of the problems in any order, where q_L and q_H are

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.

the success probabilities of the two copies on a problem. Exchanging one low-precision attempt for a high-precision one changes this by the mean gap times the fraction still open, plus a covariance between staying unsolved and the precision gap (Proposition 2). The covariance admits two readings: either mixing the two precisions has value of its own, or the value lies in sending each problem to its better precision. A test registered before the experiments decides between them. Once each problem uses its better precision, mixing precisions adds 0.69 points on average and clears the registered threshold in 3 of 12 settings (Fig. 1a). Knowing the better precision for each problem, in contrast, is worth 4.5 to 7.9 points over the better fixed schedule on the twelve settings of our main comparison.

A deployed system cannot see which precision a problem needs, but its own failures carry part of that information. A policy that sees only rejections runs a fixed sequence of attempts, so it loses nothing by running one block of the quantized copy and then one block of the BF16 copy (Proposition 4). We call this schedule RetryPrec. Its two block lengths come from an enumeration of every schedule that fits the budget, each scored by an unbiased estimate on held-out calibration problems. We show that failures lower the low-precision hazard of the survivors, as in classical frailty models, and we give the exact drift of the ratio between the two hazards, which places the switch (Theorem 1). On twelve settings at matched wall clock, RetryPrec reaches an average of 78.78 points, against 75.95 for fixed low precision, 72.38 for fixed high precision and 77.60 for the best of sixteen other strategies. When the budget charges the memory held by a second resident checkpoint, the schedule does not switch.

Our contributions are threefold. (1) A theory of retries across two precisions: the gain from exchanging one attempt, the reduction to one-switch schedules of policies that react to failures, and the drift of the hazard ratio under selection by failure. (2) RetryPrec, a calibrated one-switch schedule with an unbiased score, and a diagnostic variant that compares two switch points from only eight attempts per problem. (3) A comparison of 21 strategies under two budgets, results on nine models, eight quantization recipes and four families of tasks, and ablations that change each design choice in turn.

2 Related Work

Repeated sampling and test-time compute. Drawing several samples and keeping one that a verifier accepts raises coverage (Chen et al., 2021; Brown et al., 2024), and several works allocate such calls across models, prompts and temperatures under a budget (Snell et al., 2025; Wu et al., 2025; Chen et al., 2024a; Hassid et al., 2024; Damani et al., 2025). Closest to us, Zhang et al. (2025) choose sample counts across configurations by the coverage objective we enumerate. Self-consistency and tree search spend the budget on aggregation or branching instead (Wang et al., 2023; Yao et al., 2023), and restart strategies study when to rerun a randomized algorithm (Luby et al., 1993). We study two precisions of one model, use the joint law of their success to derive when a switch pays and after how many failures, and charge the switch and, optionally, the memory of a second resident checkpoint.

Quantization and its evaluation. GPTQ, AWQ, SmoothQuant, FP8, quantization-aware training, rotation methods and codebook methods lower the precision of the weights at a small cost in mean accuracy on standard benchmarks (Frantar et al., 2023; Lin et al., 2024; Xiao et al., 2023; Micikevicius et al., 2022; Liu et al., 2024a; Ashkboos et al., 2024; Liu et al., 2025b; Egiazarian et al., 2024; Tseng et al., 2024). Several studies systematically compare the mean accuracy of quantized models with that of the BF16 original (Jin et al., 2024; Kurtic et al., 2025; Liu et al., 2025a), and any-precision and nested schemes serve several bit widths from one set of weights (Park et al., 2024; Nair et al., 2025). We study instead how the success of the two copies varies together from problem to problem.

Cascades, routing and verifiers. FrugalGPT and related cascades call a cheap model first and escalate on rejection (Chen et al., 2024b; Yue et al., 2024; Aggarwal et al., 2024), routers pick a model before the first call (Ding et al., 2024; Ong et al., 2025), and speculative decoding drafts with a small model while keeping the output distribution of the large one (Leviathan et al., 2023; Chen et al., 2023; Li et al., 2024). Outcome and process reward models select among samples (Cobbe et al., 2021; Lightman et al., 2024; Wang et al., 2024), and self-repair and reflection retry with feedback from the failed attempt (Madaan et al., 2023; Shinn et al., 2023; Olausson et al., 2024). Each family appears in our experiments, as comparators or as systems we add the switch to. RetryPrec is closest to a cascade, with two differences: the cheap and the expensive model are two precisions of the same model, and the escalation point comes from the joint law of their success across problems.

3 RETRIES ACROSS TWO PRECISIONS

3.1 PROBLEM SETUP

Let x be a problem drawn from the distribution $\mathcal{D}$ of a task family. A configuration $b \in \{L, H\}$ is either the quantized copy L or its BF16 original H. One attempt of b on x is accepted with probability $q_b(x)$, the per-problem success probability; $\mu_b = \mathbb{E}[q_b]$ is its mean, and Var and Cov are taken over $x \sim \mathcal{D}$. A schedule σ is a sequence over $\{L, H\}$ with m entries L and n entries H, run in this order and stopped at the first accepted answer. One attempt costs c_L or c_H seconds and each change of precision costs c_{sw}, so under a budget C the feasible pairs are $\mathcal{F}_C = \{(m, n) : mc_L + nc_H + c_{\text{sw}}\mathbb{1}[m > 0, n > 0] \le C\}$. We assume (A1) independent restarts, meaning that given x the attempts are independent and a failed attempt passes nothing to the next, and (A2) a perfect verifier, which accepts an answer if and only if it is correct. Appendix A lists the symbols and terms.

Proposition 1 (Solve rate of a retry schedule)**.** *Under (A1) and (A2), every schedule with m attempts of L and n attempts of H, in any order, solves the fraction*

$$S(m, n) = 1 - \mathbb{E}\big[(1 - q_L)^m (1 - q_H)^n\big]. \tag{1}$$

Proposition 1 carries the coverage of repeated sampling (Chen et al., 2021; Brown et al., 2024) over to two configurations. The order of the attempts changes only the expected cost of a schedule, which Section 4 uses. The common practice keeps one configuration and repeats it, which gives the pure schedules $(m_{\max}, 0)$ and $(0, n_{\max})$ with the largest numbers of attempts that fit within the budget.

3.2 EXCHANGING ONE ATTEMPT

Let $w_{m,n} = (1 - q_L)^m (1 - q_H)^n$ be the probability, given x, that m attempts of L and n of H all fail. Its mean $R_{m,n} = \mathbb{E}[w_{m,n}]$ is the risk set, the fraction still open, and $h^b_{m,n} = \mathbb{E}[q_b\, w_{m,n}]/R_{m,n}$ is the hazard, the chance that one more attempt of b solves an open problem (Cox & Oakes, 1984).

Proposition 2 (Gain from exchanging one attempt)**.** *For all $m, n \ge 0$ with $R_{m,n} > 0$, exchanging one attempt of L for one attempt of H changes the solve rate by*

$$S(m, n{+}1) - S(m{+}1, n) = R_{m,n}\big(h^H_{m,n} - h^L_{m,n}\big) = R_{m,n}\,(\mu_H - \mu_L) + \text{Cov}\,\big(w_{m,n},\, q_H - q_L\big). \tag{2}$$

For $(m, n) = (1, 0)$ this reads $S(1, 1) - S(2, 0) = (1 - \mu_L)(\mu_H - \mu_L) + \text{Var}(q_L) - \text{Cov}(q_L, q_H)$.

The first term of equation 2 is the mean gap scaled by the risk set. It is all that a ranking by average accuracy can see, and it shrinks as more problems get solved. The second term is the covariance between staying unsolved and the precision gap. It is positive when the advantage of H sits on the problems that L leaves open, and by the Cauchy-Schwarz inequality it is at most half the standard deviation of $q_H - q_L$ in absolute value, so it is small when the gap is nearly the same on every problem. Appendix B proves the proposition and derives the limit of equation 2 after many failures.

The identity rests on (A1), and we checked it on sampled attempts: over 300 settings the two-attempt gain that it predicts differs from the observed one by -0.005 on average (Appendix E). We then estimated both terms on twelve model and recipe settings of the code family (Appendix Table 14). The mean gap $\mu_H - \mu_L$ is 0.023 to 0.094, and the covariance term $\text{Var}(q_L) - \text{Cov}(q_L, q_H)$ is 0.015 to 0.026, 30% to 60% of the two-attempt gain at equal attempt count. A second 4-bit recipe, or L at temperature 1.0, differs from L far less: at the reference setting their covariance terms are 0.0024 and 0.0049, against 0.0217 for H (Table 2). At equal attempt count they lead repeated attempts of L by at most 0.92 points, while drawing L or H at random per attempt leads by up to 6.5 points (Fig. 1c).

3.3 VALUE OF MIXING AND VALUE OF INFORMATION

The covariance term may mean that mixing precisions is a resource of its own, or that some problems need H and others need L, so that the value lies in choosing for each problem. To separate the two, let $S^\star_{\text{pure}}(x)$ be the largest solve rate on problem x among the feasible pure schedules and $S^\star_{\text{mix}}(x)$ the largest among all feasible pairs, both chosen with knowledge of $q_L(x)$ and $q_H(x)$. Let S_{pure} be

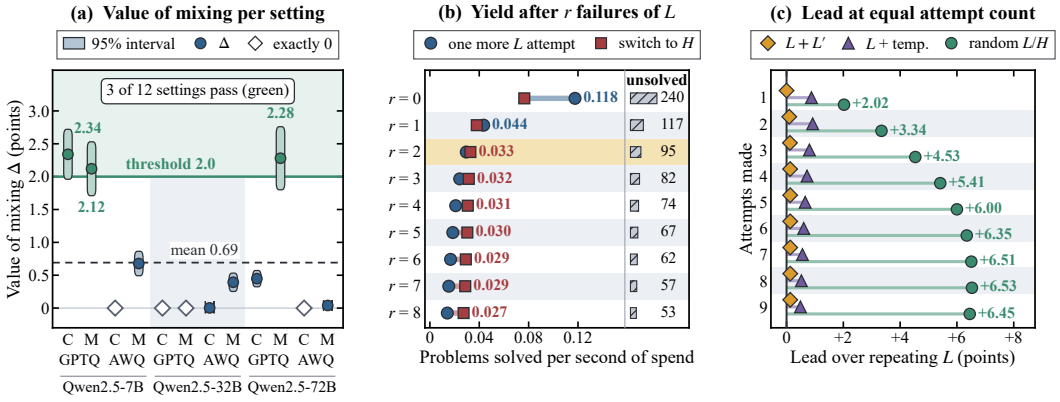


Figure 1: Retries across two precisions. (a) Value of mixing Δ, the gain of a per-problem oracle that may mix precisions over one that picks the better precision for each problem, in the twelve registered settings (C: code, M: math) with 95% intervals; the green line is the registered threshold of 2.0 points and the dashed line the mean, 0.69. (b) At the reference setting, the yield per second of one more attempt of L (circles) and of a switch to H with its charge (squares) after r failures of L; the number printed is the larger of the two, and the right column counts the problems still open out of 240. The switch is ahead from the second failure on (yellow row). (c) Lead in solve rate, in points, over the same number of attempts of L when L is mixed with a second 4-bit recipe ($L{+}L'$) or with L at temperature 1.0, and when either L or H is drawn at random for each attempt.

the better of the two population pure schedules. The value of perfect information about the better precision is $V = \mathbb{E}[S^\star_{\text{pure}}] - S_{\text{pure}}$ (Howard, 1966), and the value of mixing is $\Delta = \mathbb{E}[S^\star_{\text{mix}} - S^\star_{\text{pure}}]$.

Proposition 3 (Bound for population schedules)**.** *Every feasible pair $(m, n) \in \mathcal{F}_C$ satisfies $S(m, n) - S_{\text{pure}} \le V + \Delta$, so no population schedule can gain more than $V + \Delta$ over S_{pure}.*

On a single problem, $\log w_{m,n} = m \log(1 - q_L) + n \log(1 - q_H)$ is linear in (m, n), so over the budget its best pair is pure, and a mixed pair wins only through the rounding of the budget into a whole number of attempts (Appendix C). Before the experiments we registered one criterion: mixing counts as a resource of its own only if $\Delta \ge 2.0$ points, with a positive lower end of its 95% interval, in at least 8 of 12 settings formed by Qwen2.5-7B, 32B and 72B, GPTQ-W4 and AWQ-W4, and the code and math families. The threshold is met in 3 of the 12 settings, and Δ averages 0.69 points (Fig. 1a). On the twelve settings of Table 1, Δ averages 0.67 and V is 4.5 to 7.9 points (Appendix C). Mixing therefore adds little once the precision is chosen per problem, and most of the gain within reach comes from knowing which one each problem needs. A deployed schedule does not know $q_L(x)$ or $q_H(x)$, but it observes its own failures, and Section 4 turns these failures into that choice.

4 RetryPrec: Failure-Conditioned Precision Scheduling

4.1 Schedule and Calibration

RetryPrec runs one block of low-precision attempts and then one block of high-precision attempts. On a new problem it calls L up to $m^\star$ times and stops at the first accepted answer; if none is accepted, it switches once and calls H up to $n^\star$ times with the same rule. Both checkpoints stay resident, so a switch costs only c_{sw}, 94 ms at the reference setting. The block lengths come from N held-out calibration problems with k attempts per configuration. If $s_{b,i}$ of the k attempts of b on problem x_i are accepted, then

$$(m^\star, n^\star) = \operatorname*{arg\,max}_{(m,n)\in\mathcal{F}_C} \hat{S}(m, n), \qquad \hat{S}(m, n) = 1 - \frac{1}{N}\sum_{i=1}^{N} \hat{F}_{L,i}(m)\, \hat{F}_{H,i}(n), \tag{3}$$

where $\hat{F}_{b,i}(m) = \binom{k-s_{b,i}}{m} \Big/ \binom{k}{m}$ is the unbiased estimate of $(1 - q_b(x_i))^m$ for $m \le k$ that underlies pass@m (Chen et al., 2021). The enumeration, stated in Algorithm 1, is exact and cheap, with 13 to 37 feasible pairs per setting at Qwen2.5-32B under both budgets (Appendix Table 27), and with $k = 64$ every feasible pair in our experiments has $m, n \le k$, so the estimate applies to all of them.

Algorithm 1 RetryPrec: calibration and deployment

Require: counts $s_{b,i}$ of accepted attempts among k attempts of $b \in \{L, H\}$ on calibration problems $x_1, \ldots, x_N$; costs c_L, c_H, c_{sw}; budget C

for every pair $(m, n) \in \mathcal{F}_C$ **do**
  $\hat{S}(m, n) \leftarrow 1 - \frac{1}{N} \sum_{i=1}^{N} \hat{F}_{L,i}(m)\, \hat{F}_{H,i}(n)$, with $\hat{F}_{b,i}(m) = \binom{k - s_{b,i}}{m} / \binom{k}{m}$
$(m^\star, n^\star) \leftarrow \arg\max_{(m,n) \in \mathcal{F}_C} \hat{S}(m, n)$, ties broken toward the smaller n
Deployment on a new problem x:
for $t = 1, \ldots, m^\star$ **do run** L on x; **if** the answer is accepted **then return** it
if $m^\star > 0$ and $n^\star > 0$ **then switch** to H, at cost c_{sw}
for $t = 1, \ldots, n^\star$ **do run** H on x; **if** the answer is accepted **then return** it
return no answer

Three design choices follow from Section 3. L goes first because it solves more problems per second on the full population (Fig. 1b), and by Proposition 1 the order changes only the cost. One switch loses nothing: a policy that reacts only to failures runs a fixed sequence (Proposition 4), and blocking needs the fewest switches. The block lengths are enumerated rather than read off the first crossing of the yields, since the budget binds on the total cost and attempts are whole. A diagnostic variant uses $k = 8$ attempts per problem: unbiased estimates of the mean gap and the covariance term of Proposition 2 decide whether to switch, and the better of switching after 2 or 6 attempts of L is kept.

4.2 Theoretical Analysis

Proposition 4 (Policies that react to failures). *Under (A1) and (A2), a policy that chooses each attempt from the failures it has observed so far runs a fixed sequence, so its solve rate is $S(m, n)$ for the counts (m, n) of that sequence. Among the sequences with the same (m, n), the blocked ones change precision least and have the smallest scheduled cost. Hence $\max_{(m,n) \in \mathcal{F}_C} S(m, n)$ is the largest solve rate that any such policy can reach within a budget C on the scheduled cost.*

The enumeration of equation 3 therefore searches, through $\hat{S}$, all the policies that react to failures. The next result shows how failures change the two hazards. Fix the number r of failures of L, and write R_r, $\mathbb{E}_r$ and h_r^b for the risk set, the expectation over the problems still open, $\mathbb{E}_r[f] = \mathbb{E}[f\,(1 - q_L)^r]/R_r$, with Var_r and Cov_r defined likewise, and the hazards, all with $n = 0$. Let ν_r be the distribution whose density is proportional to $q_L(1 - q_L)^r$ on problems with $q_L > 0$, let $B_r = \mathbb{E}[q_L(1 - q_L)^r]$, and let $a_0 = \mathbb{E}[q_H \mathbb{1}\{q_L = 0\}]$ be the share of the success of H on the problems that L never solves.

Theorem 1 (Hazards after failures). *Suppose* $\mathrm{P}(0 < q_L < 1) > 0$. *(i)* $h_{r+1}^L = h_r^L - \mathrm{Var}_r(q_L)/(1 - h_r^L)$ *and* $h_{r+1}^H = h_r^H - \mathrm{Cov}_r(q_L, q_H)/(1 - h_r^L)$. *(ii) The hazard ratio* $\rho_r = h_r^H / h_r^L$ *moves as*

$$\rho_{r+1} - \rho_r = a_0 \Big(\frac{1}{B_{r+1}} - \frac{1}{B_r} \Big) - \frac{\mathrm{Cov}_{\nu_r}\big(q_H / q_L,\ q_L\big)}{1 - \mathbb{E}_{\nu_r}[q_L]}. \tag{4}$$

(iii) If $\mathrm{Cov}_{\nu_r}(q_H / q_L, q_L) \le 0$ *for every r, which is the case when $\mathbb{E}[q_H \mid q_L = t]/t$ is non-increasing in t, then ρ_r is non-decreasing, and the r with $h_r^H / (c_H + c_{\text{sw}}) > h_r^L / c_L$ form a set $\{r \ge r_0\}$. If $a_0 > 0$, then $\rho_r \to \infty$ whether or not the condition holds, and this set is not empty.*

Part (i) is the frailty effect of survival analysis (Proschan, 1963; Vaupel et al., 1979): whatever q_H does, the low-precision hazard of the survivors can only fall, while the high-precision hazard changes only through $\mathrm{Cov}_r(q_L, q_H)$ and falls when it is positive. Part (ii) gives the exact drift of the ratio of the two hazards. Its first term is never negative and, when H solves problems that L never solves, drives ρ_r to infinity as r grows; its second term is non-negative when the relative advantage of H is larger where L is weaker. Part (iii) locates the switch: under its condition the per-second comparison favors L up to some number of failures and H from then on. At the reference setting the ratio of the two yields per second rises at every failure, from 0.65 at $r = 0$ to 1.93 at $r = 8$ (Appendix Table 18).

Proposition 5 (Unbiased calibration). *Let the calibration problems be drawn independently from $\mathcal{D}$ and their attempts satisfy (A1). For $m, n \le k$, $\mathbb{E}[\hat{F}_{L,i}(m) \hat{F}_{H,i}(n) \mid x_i] = (1 - q_L(x_i))^m (1 -$*

> *$q_H(x_i))^n$, so $\mathbb{E}[\hat{S}(m,n)] = S(m,n)$. For even k, the split-half estimate of $\mathrm{Var}(q_L)$ and the cross-product estimate of $\mathrm{Cov}(q_L, q_H)$ used by the diagnostic variant are unbiased, whereas the plug-in variance of the observed success rates exceeds $\mathrm{Var}(q_L)$ by $\mathbb{E}[q_L(1-q_L)]/k$ on average.*

Unbiasedness puts every pair with $m, n \le k$ on the same footing, and since each term of $\hat{S}$ lies in $[0, 1]$, the enumeration concentrates at rate $1/\sqrt{N}$ (Appendix B). The plug-in bias matters at small k: at $k = 8$ it is 0.012 at the reference setting, more than half of the covariance term 0.0217, while the split-half estimate has a bias below 0.005 in our simulation with known truth (Appendix Table 13).

5 Experiments

5.1 Experimental Setup

Models, recipes and tasks. We use nine models: Qwen2.5-7B, 14B, 32B and 72B, Llama-3.1-8B and 70B, DeepSeek-R1-Distill-Qwen-7B and 32B, and Mistral-Small-24B (Yang et al., 2024; Grattafiori et al., 2024; Guo et al., 2025; Mistral AI Team, 2025). H is the BF16 checkpoint of each model and L one of eight recipes: GPTQ-W4, AWQ-W4, SmoothQuant-W8A8, FP8, QAT-W4, QuaRot, SpinQuant and AQLM/QuIP#. The four task families, code, math, reasoning and agent, follow the format of LiveCodeBench, AIME, GPQA and agent tasks with programmatic checks (Jain et al., 2025; Mathematical Association of America, 2026; Rein et al., 2024; Jimenez et al., 2024). For each model and family we draw 240 evaluation problems and 512 disjoint calibration problems and run 64 attempts per problem and configuration at temperature 0.7, from which $q_b(x)$ is estimated. All strategies are evaluated on the same pool of problems, so every comparison between them is paired.

Costs and budgets. Unit costs are measured on our serving stack. At the reference setting $c_L = 4.37$ s, $c_H = 7.15$ s and $c_{\mathrm{sw}} = 0.094$ s, the wall-clock budget C_{w} is 24.0 s per problem, and resident memory is 19.1 GB for L, 67.4 GB for H and 86.5 GB for both. Every attempt prefills its own prompt, since the two precisions share no KV cache. The memory-time budget is C_{w} times the resident memory of H and charges every second of an attempt with the memory held, the space-time product of memory management (Denning, 1970). The calibration is not charged (Appendix D.6).

Comparators and evaluation. Table 1 lists 21 strategies in seven groups; Appendix D.4 describes each of them. We report the solve rate in points, the percentage of problems solved, with 95% intervals of paired differences over the 240 problems from 2,000 bootstrap resamples (Efron & Tibshirani, 1993). RetryPrec uses $N = 512$ calibration problems with $k = 64$ attempts unless noted.

5.2 Main Results

Table 1 compares all strategies. At matched wall clock, RetryPrec has the highest average solve rate over the twelve settings, 78.78 points, against 75.95 for fixed L, 72.38 for fixed H and 77.60 for the strongest comparator, a switch point transferred from another setting. It is ahead of fixed L in every one of the twelve columns, by 0.42 to 5.79 points, and best or tied in 9 of the 12. At the reference setting it gains 3.43 points over the better fixed one, with a 95% paired interval of $[1.48, 5.62]$ (Appendix Table 21). Of the value of perfect information, it collects 2.49 of 6.06 points on average, or 41%, without knowing which precision any single problem needs (Appendix Table 8).

The outcome differs when the budget also charges memory. With both checkpoints resident, every second of an attempt of either precision holds 86.5 rather than 19.1 GB, so a budget that buys nineteen low-precision attempts affords at most the pair $(2, 1)$ with both resident, and the enumeration returns $n^\star = 0$ in every setting. RetryPrec then equals fixed L at 88.81 points, while diversity at one precision is slightly higher, 89.80 for L with a second temperature, which needs only one checkpoint.

We repeat the comparison of Table 1 at five budgets for code, math and reasoning (Fig. 2), against the eight other strategies of Appendix Table 24. At matched wall clock RetryPrec is ahead of the best of them in 7 of 15 settings, tied in 4 and behind the temperature baseline in 4, all at the three smaller budgets. At the two larger budgets it leads or ties in every one of the three families. Over all 30 settings of both budgets it attains the best value in 13, including 11 of the 15 at matched wall clock.

Table 1: Solve rate (points) of 21 strategies under two matched budgets. Columns: the twelve wall-clock settings (four task families by Qwen2.5-7B, 32B and 72B, with GPTQ-W4 as L), their average, the average of the same settings under the matched memory-time budget, and the peak resident memory at the reference setting. Grey rows are references and are not ranked. Bold marks the best and underline the second best solve rate in each column among the other rows; strategies that run the same pair tie (Proposition 1). More samples and best-of-n with an outcome reward model (ORM) coincide with fixed L, speculative decoding and best-of-n with a process reward model (PRM) with fixed H, and the cross-configuration allocation with $L{+}L'$ (Appendix D.4).

	Matched wall clock													Mem.-time	Peak
	Code			Math			Reasoning			Agent					
Strategy	7B	32B	72B	7B	32B	72B	7B	32B	72B	7B	32B	72B	Avg ↑	Avg ↑	GB ↓
References															
BF16, one attempt	56.85	55.43	58.60	50.56	54.60	52.89	49.98	51.60	52.38	51.82	54.87	49.22	53.23	53.23	67.38
Oracle, pure	79.26	80.05	83.53	82.55	81.78	77.74	85.29	84.73	83.15	84.25	86.79	79.11	82.35	90.26	67.38
Oracle, mixture	81.60	80.05	83.98	84.67	81.78	80.02	85.30	85.00	83.56	84.26	86.81	79.18	83.02	90.26	86.49
Fixed schedules															
Fixed L	74.78	72.12	78.13	76.16	76.30	73.05	75.26	78.71	76.81	77.07	78.74	74.32	75.95	88.81	19.11
Fixed H	67.72	71.32	74.09	70.67	72.46	65.15	77.66	75.56	73.26	76.11	80.41	64.10	72.38	72.38	67.38
Diversity at one precision															
More samples	74.78	72.12	78.13	76.16	76.30	73.05	75.26	78.71	76.81	77.07	78.74	74.32	75.95	88.81	19.11
Temperature $T{=}1.0$	75.73	74.76	78.81	75.71	**77.78**	73.30	74.68	79.41	77.56	77.63	79.22	**75.27**	76.66	89.00	19.11
$L{+}L'$ (GPTQ, AWQ)	**75.92**	72.53	78.42	76.39	76.36	73.08	75.67	78.88	77.84	78.11	78.74	75.22	76.43	89.18	19.11
L + temperature	75.73	74.76	78.84	76.28	**77.78**	73.44	75.90	79.44	77.56	77.69	79.30	**75.27**	76.83	**89.80**	19.11
Allocation between the two precisions															
Random pair	73.81	**75.55**	78.26	76.10	71.56	69.35	72.65	80.11	71.56	71.06	79.28	75.07	74.53	70.66	86.49
Fixed $m{=}\lfloor C/2c_L \rfloor$	72.35	**75.55**	**80.34**	**78.55**	75.06	72.89	79.53	79.86	**80.25**	79.31	81.88	70.66	77.19	71.64	86.49
Transferred m	75.86	**75.55**	77.90	75.48	77.50	72.89	76.36	**81.87**	**80.25**	79.31	83.18	75.07	77.60	88.05	86.49
Cross-config. allocation	**75.92**	72.53	78.42	76.39	76.36	73.08	75.67	78.88	77.84	78.11	78.74	75.22	76.43	89.18	19.11
Cascades and routing															
FrugalGPT cascade	72.35	**75.55**	77.90	75.48	76.72	69.35	**81.05**	80.42	77.43	80.13	83.18	70.66	76.69	71.41	86.49
Hybrid LLM routing	70.78	70.77	74.78	71.76	72.69	67.11	75.09	76.16	74.90	75.14	79.31	69.41	73.16	79.16	67.38
RouteLLM	71.29	71.06	76.19	73.05	73.49	69.26	76.85	77.12	74.90	76.07	79.42	68.19	73.91	80.02	67.38
Decoding and selection															
Speculative decoding	67.72	71.32	74.09	70.67	72.46	65.15	77.66	75.56	73.26	76.11	80.41	64.10	72.38	72.38	67.38
Best-of-n, ORM	74.78	72.12	78.13	76.16	76.30	73.05	75.26	78.71	76.81	77.07	78.74	74.32	75.95	88.81	19.11
Best-of-n, PRM	67.72	71.32	74.09	70.67	72.46	65.15	77.66	75.56	73.26	76.11	80.41	64.10	72.38	72.38	67.38
Ours															
RetryPrec	75.86	**75.55**	**80.34**	**78.55**	76.72	**75.27**	**81.05**	**81.87**	**80.25**	**81.18**	**83.69**	75.07	**78.78**	88.81	86.49
RetryPrec, diagnostic	72.35	**75.55**	77.90	75.48	76.72	**75.27**	**81.05**	80.42	77.43	80.13	83.18	70.66	77.18	88.81	86.49

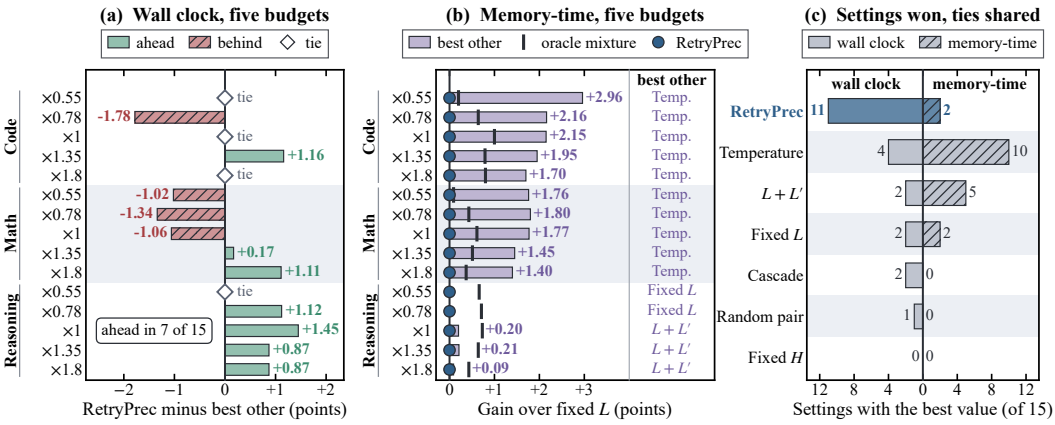


Figure 2: Comparison at equal cost (Qwen2.5-32B, GPTQ-W4; code, math and reasoning at five budgets, 0.55 to 1.8 times the nominal one). (a) RetryPrec minus the best of the eight other strategies of Appendix Table 24 at matched wall clock. (b) At matched memory-time, the gain over fixed L of the best other strategy (bar) and of the per-problem L/H oracle mixture (tick); RetryPrec (dot) returns fixed L in every setting, and the right column names the best other strategy. (c) Number of settings, out of 15 on each side, in which each strategy is the best; ties count for every tied strategy, and fixed L and fixed H also stand for more samples and speculative decoding.

Table 2: Ablation studies at Qwen2.5-32B with GPTQ-W4 as L, matched wall clock. Solve rate in points per family, the average, and its change from the full method; each row changes one design choice. The last column is the covariance term $\mathrm{Var}(q_L) - \mathrm{Cov}(q_L, q_{b'})$ at the reference setting for the partner b' of L; it is 0.0217 for H.

Variant	Code	Math	Reas.	Agent	Avg ↑	Change	Cov. term
RetryPrec (full)	75.55	76.72	81.87	83.69	79.46	–	0.0217
Switch point							
random feasible pair	75.55	71.56	80.11	79.28	76.62	−2.84	
fixed $m = \lfloor C/2c_L \rfloor$	75.55	75.06	79.86	81.88	78.09	−1.37	
diagnostic variant (k=8)	75.55	76.72	80.42	83.18	78.97	−0.49	
Partner of L							
q_H replaced by q_L	72.12	76.30	78.71	78.74	76.47	−2.99	0
second 4-bit recipe (L+L')	72.53	76.36	78.88	78.74	76.63	−2.83	0.0024
L at T=1.0 (L + temp.)	74.76	77.78	79.44	79.30	77.82	−1.64	0.0049
Order and switches							
H block first	75.47	76.58	81.71	83.55	79.33	−0.13	
interleaved L and H	75.42	76.72	81.87	83.69	79.43	−0.03	
two switches allowed	75.61	76.80	81.93	83.76	79.53	+0.07	
Serving							
reload H from host memory	73.52	75.06	80.11	81.20	77.47	−1.99	
with prefix caching	76.14	76.72	82.69	82.73	79.57	+0.11	
visible-test verifier	75.23	76.39	81.29	83.04	78.99	−0.47	
Calibration							
$k = 8$ attempts per problem	75.55	76.72	81.87	83.69	79.46	0.00	
$N = 128$ problems	75.52	76.69	81.84	83.62	79.42	−0.04	
$N = 64$ problems	75.55	76.72	78.71	83.69	78.67	−0.79	

5.3 Ablation Studies

Each row of Table 2 changes one design choice of RetryPrec at Qwen2.5-32B with GPTQ-W4 and matched wall clock; Appendices F and G give each variant per family, most with its pair and costs.

Calibrating the switch point matters. A feasible pair drawn at random per family loses 2.84 points on average, a fixed switch at half the budget loses 1.37, and the diagnostic variant 0.49. The drawn pair loses on math, reasoning and agent, where it keeps a single attempt of H (Appendix Table 28).

The high-precision copy is the best partner. With q_H replaced by q_L on every problem and the schedule recalibrated, the enumeration returns the pure low-precision schedule and the gain over fixed L is gone. A second 4-bit recipe or L at temperature 1.0 as the partner recovers only part of it, 76.63 and 77.82 points against 79.46, in line with their much smaller covariance terms in Table 2.

Low precision first and blocking save cost. With the same pair, running the high-precision block first raises the expected cost by 1.5 to 3.4 s, and interleaving adds more switch charges and up to 2.7 s (Appendix Tables 30 and 31). The solve rate, independent of the order in expectation (Proposition 1), moves by at most 0.16 points on sampled attempts, and by at most 0.08 points with a second switch.

Keeping H resident matters. Reloading H from host memory at 24 GB/s instead of keeping it resident shortens the high-precision block and costs 1.99 points on average at matched wall clock.

A small calibration suffices. Eight attempts per calibration problem give the same schedule as 64 attempts, and 128 calibration problems come within 0.04 points of 512; with 64 problems the reasoning family falls back to the pure low-precision schedule and the average drops by 0.79 points.

The schedule composes with serving methods. Prefix caching lowers the unit costs of attempts, the enumeration moves the switch point with them, and the average goes up by 0.11 points; with a visible-test verifier that accepts 0.73% of wrong answers, the average falls by 0.47 points (Appendix G.4).

5.4 Analysis

Other models and recipes. Across the nine models the four-family average of RetryPrec lies between 73.33 and 79.46 points, and across eight recipes between 76.27 and 79.46 (Appendix

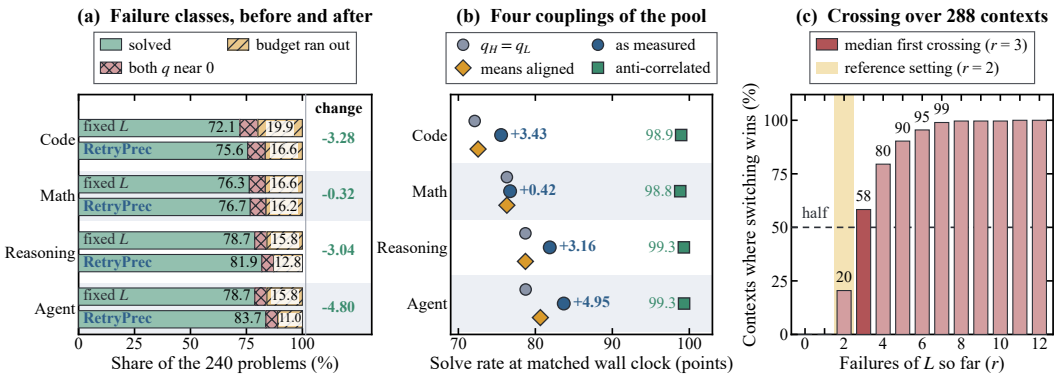


Figure 3: Where the gain comes from. (a) Share of the 240 problems per family that are solved, have both q near zero, or ran out of budget, under fixed L (upper) and RetryPrec (lower), with the change in the last class. (b) Solve rate of RetryPrec at matched wall clock under four couplings of the pool, q_H replaced by q_L, means aligned, as measured and anti-correlated failures; blue: gain as measured over the first; green: anti-correlated solve rate. (c) Share of the 288 contexts in which the switch to H has the higher yield per second after r failures.

Table 25). Against the best other strategy of each of the 36 model and family settings, chosen separately for each setting, the 95% paired interval covers zero in 33 of them (Appendix Table 26).

The gain follows the coupling. As in the ablation, with q_H set equal to q_L the enumeration keeps the pure low-precision schedule and gains nothing; with the two means aligned the gain shrinks to 0 to 1.9 points; with the failures made anti-correlated the recalibrated schedule solves 98.8 to 99.3 points (Fig. 3b). In every family the gain follows the order of the covariance term of Proposition 2.

Crossing point across contexts. Over 288 model, recipe and family contexts the switch overtakes one more low-precision attempt after two to eleven failures, with a median of three (Fig. 3c). The share of contexts in which the switch is ahead equals the share that have crossed at least once, at every r up to 12, so no context crosses back, the single-crossing pattern that Theorem 1 describes.

Cost. At the reference setting RetryPrec spends 11.49 s per problem on average when it stops at the first accepted answer, against 11.05 s for fixed L and 12.79 s for fixed H (Appendix Table 21). A calibration with 128 problems and 8 attempts each costs 11,796 s on code, or 2.36 s per served problem over 5,000 problems, and matches the full calibration on code and math (Appendix D.6).

Which problems gain. Compared with fixed L, RetryPrec shrinks the failure class of problems whose budget ran out by 0.32 to 4.80 points in each of the four families, while the class on which both precisions are near zero moves by at most 0.15 points (Fig. 3a). The gain comes from problems that stay open under fixed L even though H can solve them, the group that failures select. At the reference setting, 47 of the 240 problems have $q_L < 0.2$ and q_H from 0.3 to 0.6 (Appendix Table 16).

6 Conclusion

We studied retries across two precisions of one language model. A failed attempt is a selection event, so the value of a switch depends on the joint law of the two per-problem success probabilities, of which their means are one part. RetryPrec acts on this with one calibrated switch from low to high precision, and at matched wall clock it gives the best average solve rate among the deployable strategies of Table 1. It requires the verdicts of the verifier, a small calibration set and both of the checkpoints resident. Most of what a population schedule can gain lies in choosing a precision for each problem, and a schedule that sees only its failures collects 41% of it on average in our settings.

Limitations. The analysis assumes independent restarts and a verifier that accepts only correct answers; retries with feedback and leaky verifiers enter our experiments as controls or as host systems. The schedule takes both configurations as given, with their sampling temperature fixed (Appendix G), and when the budget charges the memory of a second resident checkpoint, it keeps to low precision.

Reproducibility Statement

RetryPrec is specified by equation 3 and Algorithm 1, and the estimates of the diagnostic variant by equation 19. Appendix A lists the symbols, Appendix B proves every statement, and Appendix D describes the problem pools, the accounting of costs, the calibration and the comparators. Appendices F and G report the full results and ablations. The pools of per-problem success probabilities, the measured unit costs and the evaluation code will all be released with the camera-ready version.

AI Use Statement

Generative AI tools were used only to edit the wording of text written by the authors, to search for related work and check bibliographic details, and to write code for plots, tables and typesetting. They were not used for any task that the ICLR 2027 AI policy requires authors to disclose: developing theory, formulating or proving mathematical claims, proposing or refining hypotheses, designing, implementing or running the method or experiments, generating, cleaning or analyzing data, translating text, or interpreting results. The models in our experiments are objects of study, not writing aids. The authors reviewed all AI-assisted work and take full responsibility for the content of this paper.

References

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 12375–12396, 2023.

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, et al. AutoMix: Automatically mixing language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 131000–131034, 2024.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, et al. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 100213–100240, 2024. arXiv:2404.00456.

J. Martin Bland and Douglas G. Altman. Statistical methods for assessing agreement between two methods of clinical measurement. *The Lancet*, 327(8476):307–310, 1986.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

Lingjiao Chen, Jared Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. Are more LM calls all you need? Towards the scaling properties of compound AI systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 45767–45790, 2024a. arXiv:2403.02419.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research (TMLR)*, 2024b. arXiv:2305.05176.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

D. R. Cox and D. Oakes. *Analysis of Survival Data*. Monographs on Statistics and Applied Probability. Chapman and Hall, London, 1984.

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. Learning how hard to think: Input-adaptive allocation of LM computation. In *International Conference on Learning Representations (ICLR)*, pp. 102783–102802, 2025.

Peter J. Denning. Virtual memory. *ACM Computing Surveys*, 2(3):153–189, 1970.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations (ICLR)*, pp. 41348–41366, 2024.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall, New York, 1993.

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12284–12303, 2024.

J. D. Esary, F. Proschan, and D. W. Walkup. Association of random variables, with applications. *The Annals of Mathematical Statistics*, 38(5):1466–1474, 1967.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations (ICLR)*, 2023. The arXiv version (arXiv:2210.17323) is titled "GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers".

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. Preprint version: "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning", arXiv:2501.12948.

Michael Hassid, Tal Remez, Jonas Gehring, Roy Schwartz, and Yossi Adi. The larger the better? Improved LLM code-generation via budget reallocation. In *First Conference on Language Modeling (COLM)*, 2024.

Wassily Hoeffding. A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293–325, 1948.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Ronald A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26, 1966.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, et al. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations (ICLR)*, pp. 58791–58831, 2025. arXiv:2403.07974.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, pp. 54107–54157, 2024.

Renren Jin, Jiangcun Du, Wuwei Huang, Wei Liu, Jian Luan, Bin Wang, and Deyi Xiong. A comprehensive evaluation of quantization strategies for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12186–12215, 2024.

Eldar Kurtic, Alexandre Noll Marques, Shubhra Pandit, Mark Kurtz, and Dan Alistarh. "Give me BF16 or give me death"? Accuracy-performance trade-offs in LLM quantization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pp. 26872–26886, 2025. arXiv:2411.02355.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19274–19286, 2023.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. EAGLE: Speculative sampling requires rethinking feature uncertainty. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28935–28948, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, et al. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, pp. 39578–39601, 2024.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, et al. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 6, pp. 87–100, 2024. arXiv:2306.00978.

Ruikang Liu, Yuxuan Sun, Manyi Zhang, Haoli Bai, Xianzhi Yu, Tiezheng Yu, Chun Yuan, and Lu Hou. Quantization hurts reasoning? An empirical study on quantized reasoning models. In *Conference on Language Modeling (COLM)*, 2025a. arXiv:2504.04823.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, et al. LLM-QAT: Data-free quantization aware training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 467–484, 2024a.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, et al. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations (ICLR)*, pp. 92009–92032, 2025b. arXiv:2405.16406.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32332–32344, 2024b.

Michael Luby, Alistair Sinclair, and David Zuckerman. Optimal speedup of Las Vegas algorithms. *Information Processing Letters*, 47(4):173–180, 1993.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, et al. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 46534–46594, 2023.

Mathematical Association of America. MAA invitational competitions. https://maa.org/maa-invitational-competitions/, 2026. Section "American Invitational Mathematics Examination (AIME)". Accessed September 2026.

Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, et al. FP8 formats for deep learning. *arXiv preprint arXiv:2209.05433*, 2022.

Mistral AI Team. Mistral Small 3. https://mistral.ai/news/mistral-small-3/, January 2025. Blog post, 30 January 2025.

L. G. Mitten. An analytic solution to the least cost testing sequence problem. *The Journal of Industrial Engineering*, 11(1):17, 1960.

Pranav Ajit Nair, Puranjay Datta, Jeff Dean, Prateek Jain, and Aditya Kusupati. Matryoshka quantization. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 45484–45506, 2025.

Theo X. Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. Is self-repair a silver bullet for code generation? In *International Conference on Learning Representations (ICLR)*, pp. 36545–36593, 2024.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs with preference data. In *International Conference on Learning Representations (ICLR)*, pp. 34433–34448, 2025. arXiv:2406.18665.

Yeonhong Park, Jake Hyun, Sanglyul Cho, Bonggeun Sim, and Jae W. Lee. Any-Precision LLM: Low-cost deployment of multiple, different-sized LLMs. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39682–39701, 2024.

George R. Price. Selection and covariance. *Nature*, 227(5257):520–521, 1970. doi: 10.1038/227520a0.

Frank Proschan. Theoretical explanation of observed decreasing failure rate. *Technometrics*, 5(3): 375–383, 1963.

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. arXiv:2311.12022.

Moshe Shaked and J. George Shanthikumar. *Stochastic Orders*. Springer Series in Statistics. Springer, New York, 2007.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 8634–8652, 2023.

Herbert A. Simon and Joseph B. Kadane. Optimal problem-solving search: All-or-none solutions. *Artificial Intelligence*, 6(3):235–247, 1975.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations (ICLR)*, pp. 10131–10165, 2025. arXiv:2408.03314.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: Even better LLM quantization with Hadamard incoherence and lattice codebooks. In *International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48630–48656, 2024.

James W. Vaupel, Kenneth G. Manton, and Eric Stallard. The impact of heterogeneity in individual frailty on the dynamics of mortality. *Demography*, 16(3):439–454, 1979.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, et al. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pp. 9426–9439, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *International Conference on Learning Representations (ICLR)*, pp. 55824–55845, 2025. arXiv:2408.00724.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38087–38099, 2023.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 11809–11822, 2023.

Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thought representations for cost-efficient reasoning. In *International Conference on Learning Representations (ICLR)*, pp. 21691–21728, 2024.

Kexun Zhang, Shang Zhou, Danqing Wang, William Yang Wang, and Lei Li. Scaling LLM inference efficiently with optimized sample compute allocation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), Volume 1: Long Papers*, pp. 7959–7973, 2025. arXiv:2410.22480.

APPENDIX

A Notation **16**

B Proofs **17**

B.1 Proof of Proposition 1 and the Cost of an Order . . . 17
B.2 Proof of Proposition 2 and the Limit of Many Failures . . . 18
B.3 Proof of Proposition 4 . . . 20
B.4 Proof of Theorem 1 . . . 20
B.5 Proof of Proposition 5 and Concentration of the Calibration . . . 21
B.6 Proof of Proposition 3 and a Decomposition of the Gain . . . 22
B.7 Prices Under the Memory-Time Budget . . . 23

C The Registered Criterion **23**

D Method and Experimental Details **25**

D.1 Diagnostic Variant . . . 25
D.2 Pools, Tasks and Models . . . 25
D.3 Cost Accounting and Budgets . . . 25
D.4 Strategies . . . 26
D.5 Validity Checks . . . 27
D.6 Cost of the Calibration . . . 27

E Checks of the Identity, the Estimators and the Pools **28**

E.1 The Identity on Bernoulli Attempts . . . 28
E.2 Simulation Check of the Estimators . . . 28
E.3 Two-Attempt Diagnostics per Setting . . . 29
E.4 Joint Law of the Pools . . . 29
E.5 Survivors After Failures . . . 30
E.6 Numerical Checks of the Exchange Identity and the Diagnostic Rule . . . 32

F Full Results **33**

F.1 The Reference Setting . . . 33
F.2 All Strategies at Qwen2.5-32B . . . 33
F.3 Equal-Cost Comparison and Budget Sweep . . . 33
F.4 Models, Recipes, Verifiers and Retry Mechanisms . . . 36
F.5 Crossing Point Across Contexts . . . 36

G Complete Ablation Studies **37**

G.1 Switch Point . . . 37
G.2 Partner of the Low-Precision Copy . . . 38
G.3 Order and Number of Switches . . . 38
G.4 Residency and Serving Methods . . . 40
G.5 Calibration . . . 41
G.6 Failure Classes . . . 41

A NOTATION

Table 3 lists the main symbols of the paper with the place where each is first defined, and Table 4 fixes the meaning of the terms used in a technical sense. Probabilities, means, variances and covariances are taken over the problems $x \sim \mathcal{D}$ unless a subscript names some other law, and the solve rates of the strategies are reported in points throughout, that is, in percent of all the problems that are solved.

Table 3: Symbols, with their meaning and the place of their first definition.

Symbol	Meaning	Defined in
x, $\mathcal{D}$	a problem, and the distribution of the problems of a task family	Section 3.1
$b \in \{L, H\}$	a configuration: the quantized copy L or the BF16 original H	Section 3.1
$q_b(x)$, μ_b	probability that one attempt of b on x is accepted, and its mean $\mathbb{E}[q_b]$	Section 3.1
σ, (m, n)	a schedule, and its numbers of attempts of L and of H	Section 3.1
c_L, c_H, c_{sw}	cost in seconds of one attempt of L, of H, and of one change of precision	Section 3.1
C, $\mathcal{F}_C$	a budget, and the pairs (m, n) that fit it	Section 3.1
$S(m, n)$	solve rate of every schedule with m attempts of L and n of H	Eq. (1)
$S(\sigma)$	solve rate of a schedule σ	Appendix B.1
$m_{\max}$, $n_{\max}$	largest number of attempts of L alone, or of H alone, within C	Section 3.1
$w_{m,n}$, $R_{m,n}$	probability, given x, that m attempts of L and n of H all fail, and its mean, the risk set	Section 3.2
$h^b_{m,n}$	conditional hazard: chance that one more attempt of b solves an open problem	Section 3.2
r	number of failures of L	Section 4.2
$\mathbb{E}_r$, Var_r, Cov_r	moments over the problems still open after r failures of L	Section 4.2
h^b_r, R_r	hazard of b and risk set after r failures of L, that is $h^b_{r,0}$ and $R_{r,0}$	Section 4.2
ρ_r, r_0	hazard ratio h^H_r / h^L_r, and the first r from which a switch has the higher yield per second	Theorem 1
τ	price ratio $(c_H + c_{\mathrm{sw}})/c_L$ with which ρ_r is compared	Appendix B.4
ν_r, B_r, a_0	law with density $\propto q_L(1-q_L)^r$ on $\{q_L > 0\}$, its normalizer $\mathbb{E}[q_L(1-q_L)^r]$, and $\mathbb{E}[q_H \mathbb{1}\{q_L = 0\}]$	Section 4.2
A_r, g, $\bar{g}$	$\mathbb{E}[q_H(1-q_L)^r]$, the ratio q_H/q_L, and $\bar{g}(t) = \mathbb{E}[q_H \mid q_L = t]/t$	Appendix B.4
$S^\star_{\mathrm{pure}}(x)$, $S^\star_{\mathrm{mix}}(x)$	largest solve probability on problem x over the feasible pure pairs, and over all feasible pairs, given $q_L(x)$ and $q_H(x)$	Section 3.3
S_{pure}	solve rate of the better of the two pure schedules $(m_{\max}, 0)$ and $(0, n_{\max})$	Section 3.3
V, Δ	value of perfect information $\mathbb{E}[S^\star_{\mathrm{pure}}] - S_{\mathrm{pure}}$, and value of mixing $\mathbb{E}[S^\star_{\mathrm{mix}} - S^\star_{\mathrm{pure}}]$	Section 3.3
G	gain $S(m^\star, n^\star) - S_{\mathrm{pure}}$ of the calibrated schedule	Appendix B.6
N, k	number of calibration problems, and attempts per problem and configuration	Section 4.1
$s_{b,i}$	accepted attempts of b among the k attempts on calibration problem x_i	Section 4.1
$\hat{F}_{b,i}(m)$, $\hat{S}(m, n)$	unbiased estimate of $(1 - q_b(x_i))^m$, and the calibration estimate of $S(m, n)$	Eq. (3)
$(m^\star, n^\star)$	the pair chosen by RetryPrec	Eq. (3)
$\hat{p}_{b,i}$, $\hat{p}^A_{b,i}$, $\hat{p}^B_{b,i}$	observed success rate of b on x_i, and on two disjoint halves of its attempts	Appendix B.5
$\widehat{\mathrm{Var}}(q_L)$	split-half estimate of $\mathrm{Var}(q_L)$, used by the diagnostic variant	Eq. (19)
$\widehat{\mathrm{Cov}}(q_L, q_H)$	unbiased estimate of $\mathrm{Cov}(q_L, q_H)$, used by the diagnostic variant	Eq. (19)
ε_N, δ	deviation bound of the calibration, and the probability that it fails	Appendix B.5
W, h^b_W	probability that the attempts before position t all fail, and the hazards it defines	Appendix B.1
C_{w}	wall-clock budget (s)	Section 5.1
C_{m}	memory-time budget $C_{\mathrm{w}} M_H$ (GB·s)	Appendix B.7
M_L, M_H, κ	resident memory of L and of H (GB), and the memory ratio $(M_L + M_H)/M_L$	Appendix B.7
$\bar{m}$	number of attempts of L alone that fit the memory-time budget	Appendix B.7

Table 4: Terms used in a fixed technical sense.

Term	Meaning
independent restarts	given the problem, attempts are independent, and nothing from a failed attempt is passed to the next one; assumption (A1)
perfect verifier	accepts an answer if and only if it is correct; assumption (A2). The visible-test verifier, a control, accepts 0.73% of wrong answers on average
pure, mixed schedule	a schedule is pure if all its attempts use one precision, and mixed otherwise
population, per-problem	a population schedule uses one pair (m, n) for every problem; a per-problem oracle chooses the pair for each problem from its true $q_b(x)$
setting	one combination of model, recipe, task family and budget
context	one combination of model, recipe and task family; $9 \times 8 \times 4 = 288$ in all
reference setting	Qwen2.5-32B, GPTQ-W4 as L, the code family, matched wall clock
pool	for each context, the pairs $(q_L(x), q_H(x))$ of 240 evaluation problems, each estimated from 64 attempts per configuration
calibration problems	512 problems per context, disjoint from the pool, with 64 attempts per configuration, from which RetryPrec chooses $(m^\star, n^\star)$
resident, reload	a checkpoint is resident when it stays in accelerator memory; a reload copies it from host memory at 24 GB/s
memory-time	gigabytes held multiplied by the seconds they are held, the space-time product of memory management (Denning, 1970)
covariance term	the term $\mathrm{Cov}(w_{m,n}, q_H - q_L)$ of Proposition 2, which for two attempts is $\mathrm{Var}(q_L) - \mathrm{Cov}(q_L, q_H)$
yield per second	hazard of the next attempt divided by its cost, h_r^L/c_L for L and $h_r^H/(c_H + c_{\mathrm{sw}})$ for a switch to H
diagnostic variant	RetryPrec with the enumeration replaced by the estimates of equation 19, which decide whether to switch, and a choice between switching after 2 or after 6 attempts of L, all from $k = 8$ attempts

B Proofs

This appendix proves every statement of the main text and adds five further results that the experiments use: the expected cost of an order (Lemma 1), the exchange of two adjacent attempts (Proposition 6), the limit of the exchange identity after many failures of L (Corollary 1), the concentration of the calibration (Proposition 7), and the attempts that a memory-time budget buys (Proposition 8). Where it helps the reader, a proof is followed by a check on the numbers of the reference setting, or on a small example with only two types of problems for which every number can be worked out by hand.

B.1 Proof of Proposition 1 and the Cost of an Order

Proof of Proposition 1. Fix a problem x and a schedule $\sigma = (\sigma_1, \dots, \sigma_{m+n})$. Under (A2) the schedule solves x exactly when one of its attempts is accepted. Stopping at the first accepted answer does not change this event, because every attempt that the stopped run skips would have come after an accepted one. Under (A1) the outcomes are independent given x, so the chance that the schedule fails factorizes and can be regrouped by configuration, and averaging over $x \sim \mathcal{D}$ gives

$$
\begin{aligned}
S(\sigma) &= 1 - \mathbb{E}\Big[\,\mathrm{P}\left(\text{no attempt of } \sigma \text{ is accepted} \;\middle|\; x\right)\Big] = 1 - \mathbb{E}\Big[\prod_{t=1}^{m+n}\big(1 - q_{\sigma_t}(x)\big)\Big] \\
&= 1 - \mathbb{E}\Big[\prod_{t:\,\sigma_t = L}\big(1 - q_L(x)\big)\prod_{t:\,\sigma_t = H}\big(1 - q_H(x)\big)\Big] \\
&= 1 - \mathbb{E}\Big[\big(1 - q_L(x)\big)^m\big(1 - q_H(x)\big)^n\Big] = S(m, n). \qquad (5)
\end{aligned}
$$

The right-hand side depends on σ only through (m, n), which proves the claim for every order. □

The proof uses only (A1) and (A2). Retries with feedback break (A1), since the second attempt then depends on the content of the first failure, and a verifier that accepts some wrong answers breaks (A2),

because a false acceptance stops the schedule. Both appear in the experiments as controls (Tables 25 and 33), and a self-repair loop also serves as a host system to which the switch is added (Table 34).

Lemma 1 (Expected cost of an order)**.** *Under (A1), when the schedule stops at the first accepted answer, the expected cost of σ is*

$$\mathbb{E}[\mathrm{cost}(\sigma)] = \sum_{t=1}^{m+n} \Big(c_{\sigma_t} + c_{\mathrm{sw}} \mathbb{1}[t \geq 2,\ \sigma_t \neq \sigma_{t-1}] \Big)\, \mathbb{E}\Big[\prod_{l<t} \big(1 - q_{\sigma_l}\big) \Big]. \tag{6}$$

Proof. Attempt t runs exactly when the attempts $1, \ldots, t-1$ all fail, which given x has probability $\prod_{l<t}(1 - q_{\sigma_l}(x))$ by (A1). It costs c_{σ_t}, plus c_{sw} when it changes the precision. Weighting each charge by the chance that its attempt runs, summing over t and averaging over x gives equation 6. □

Proposition 6 (Exchange of two adjacent attempts)**.** *Let σ have H at position t and L at position $t{+}1$, and let σ' swap the two. Write $W = \prod_{l<t}(1 - q_{\sigma_l})$ and $h^b_W = \mathbb{E}[q_b W]/\mathbb{E}[W]$ for the hazards of the problems still open before position t. Without switch charges,*

$$\mathbb{E}[\mathrm{cost}(\sigma')] - \mathbb{E}[\mathrm{cost}(\sigma)] = \mathbb{E}[W]\, \big(c_L\, h^H_W - c_H\, h^L_W \big), \tag{7}$$

so placing L before H is no more expensive exactly when $h^L_W / c_L \geq h^H_W / c_H$. For two attempts and with the switch charge, $\mathbb{E}[\mathrm{cost}(LH)] - \mathbb{E}[\mathrm{cost}(HL)] = c_L \mu_H - c_H \mu_L + c_{\mathrm{sw}}(\mu_H - \mu_L)$.

Proof. Without switch charges the two orders run the same attempts before position t and, from position $t{+}2$ on, have the same product $\prod_{l \leq t+1}(1 - q_{\sigma_l})$ in equation 6, so only the terms t and $t{+}1$ differ. For σ they contribute $c_H \mathbb{E}[W] + c_L \mathbb{E}[W(1 - q_H)]$, and for σ' they contribute $c_L \mathbb{E}[W] + c_H \mathbb{E}[W(1 - q_L)]$. Hence

$$\begin{aligned}
&\mathbb{E}[\mathrm{cost}(\sigma')] - \mathbb{E}[\mathrm{cost}(\sigma)] \\
&\quad = \Big(c_L\, \mathbb{E}[W] + c_H\, \mathbb{E}\big[W(1 - q_L)\big] \Big) - \Big(c_H\, \mathbb{E}[W] + c_L\, \mathbb{E}\big[W(1 - q_H)\big] \Big) \\
&\quad = c_L \mathbb{E}[W] + c_H \mathbb{E}[W] - c_H \mathbb{E}[W q_L] - c_H \mathbb{E}[W] - c_L \mathbb{E}[W] + c_L \mathbb{E}[W q_H] \\
&\quad = c_L\, \mathbb{E}[W q_H] - c_H\, \mathbb{E}[W q_L] = \mathbb{E}[W]\, \big(c_L\, h^H_W - c_H\, h^L_W \big).
\end{aligned} \tag{8}$$

For two attempts, $W = 1$ and the switch is paid when the second attempt is reached, with probability $1 - \mu_L$ under LH and $1 - \mu_H$ under HL:

$$\begin{aligned}
\mathbb{E}[\mathrm{cost}(LH)] - \mathbb{E}[\mathrm{cost}(HL)] &= \Big(c_L + (c_H + c_{\mathrm{sw}})(1 - \mu_L) \Big) - \Big(c_H + (c_L + c_{\mathrm{sw}})(1 - \mu_H) \Big) \\
&= c_L \mu_H - c_H \mu_L + c_{\mathrm{sw}}(\mu_H - \mu_L).
\end{aligned}$$

□

Proposition 6 is the population form of the rule that orders independent tests by their success probability per unit of cost (Mitten, 1960; Simon & Kadane, 1975). Here the probabilities are the hazards of the open problems, and these move with every failure, so the switch point has to follow the failures. At the reference setting, $\mu_L = 0.514$ and $\mu_H = 0.554$ (Table 18), $c_L = 4.37$ s, $c_H = 7.15$ s and $c_{\mathrm{sw}} = 0.094$ s, so $\mathbb{E}[\mathrm{cost}(LH)] = 4.37 + 7.244 \times 0.486 = 7.89$ s and $\mathbb{E}[\mathrm{cost}(HL)] = 7.15 + 4.464 \times 0.446 = 9.14$ s. Low precision first saves $c_H \mu_L - c_L \mu_H = 1.254$ s of attempt cost and pays 0.004 s more in switch charges, a net saving of 1.25 s; for the calibrated pair $(2, 2)$ the expected costs are 11.49 and 13.10 s (Table 30), and low precision first again saves 1.61 s.

B.2 Proof of Proposition 2 and the Limit of Many Failures

Proof of Proposition 2. By Proposition 1, with $w_{m,n} = (1 - q_L)^m (1 - q_H)^n$,

$$\begin{aligned}
S(m, n{+}1) - S(m{+}1, n) &= \mathbb{E}\big[(1 - q_L)^{m+1}(1 - q_H)^n\big] - \mathbb{E}\big[(1 - q_L)^m (1 - q_H)^{n+1}\big] \\
&= \mathbb{E}\Big[w_{m,n} \big((1 - q_L) - (1 - q_H) \big) \Big] = \mathbb{E}\big[w_{m,n}\, (q_H - q_L) \big] \\
&= \mathbb{E}\big[w_{m,n}\, q_H \big] - \mathbb{E}\big[w_{m,n}\, q_L \big] = R_{m,n} \big(h^H_{m,n} - h^L_{m,n} \big),
\end{aligned} \tag{9}$$

where the last step divides and multiplies by $R_{m,n} = \mathbb{E}[w_{m,n}] > 0$. For the second form, write the mean of a product as the product of the means plus the covariance:

$$\begin{aligned}\mathbb{E}\big[w_{m,n}(q_H - q_L)\big] &= \mathbb{E}[w_{m,n}]\,\mathbb{E}[q_H - q_L] + \operatorname{Cov}\big(w_{m,n},\, q_H - q_L\big) \\ &= R_{m,n}(\mu_H - \mu_L) + \operatorname{Cov}\big(w_{m,n},\, q_H - q_L\big). \end{aligned} \tag{10}$$

For two attempts, $w_{1,0} = 1 - q_L$ and $R_{1,0} = 1 - \mu_L$, and the covariance splits by bilinearity:

$$\begin{aligned} S(1,1) - S(2,0) &= (1-\mu_L)(\mu_H - \mu_L) + \operatorname{Cov}\big(1 - q_L,\; q_H - q_L\big) \\ &= (1-\mu_L)(\mu_H - \mu_L) - \operatorname{Cov}(q_L, q_H) + \operatorname{Cov}(q_L, q_L) \\ &= (1-\mu_L)(\mu_H - \mu_L) + \operatorname{Var}(q_L) - \operatorname{Cov}(q_L, q_H). \end{aligned}$$ □

Size of the covariance term. Since $0 \le w_{m,n} \le 1$, we have $\mathbb{E}[w_{m,n}^2] \le \mathbb{E}[w_{m,n}]$ and hence $\operatorname{Var}(w_{m,n}) \le R_{m,n}(1 - R_{m,n}) \le \frac{1}{4}$. The Cauchy-Schwarz inequality then gives

$$\big|\operatorname{Cov}(w_{m,n}, q_H - q_L)\big| \le \sqrt{R_{m,n}(1 - R_{m,n})}\,\operatorname{sd}(q_H - q_L) \le \tfrac{1}{2}\operatorname{sd}(q_H - q_L), \tag{11}$$

which is the bound quoted in Section 3.2. Dividing equation 2 by $R_{m,n}$ shows how the balance between the two terms moves with the risk set. Per open problem the mean term stays equal to $\mu_H - \mu_L$, while the covariance term may grow like $\sqrt{(1 - R_{m,n})/R_{m,n}}$ when the risk set shrinks, so after many failures the second term can decide the exchange on its own, whatever the mean gap.

Corollary 1 (Many failures). *Suppose* $\mathrm{P}(q_L = 0,\, q_H < 1) > 0$. *For fixed* n, *as* $m \to \infty$,

$$h^L_{m,n} \to 0, \qquad h^H_{m,n} \to \frac{\mathbb{E}\big[q_H(1-q_H)^n \mathbb{1}\{q_L = 0\}\big]}{\mathbb{E}\big[(1-q_H)^n \mathbb{1}\{q_L = 0\}\big]}, \tag{12}$$

so, per open problem, the value of exchanging one attempt of L for one attempt of H converges to the hazard of H among the problems that L never solves, the only ones left open in the limit.

Proof. As $m \to \infty$, $(1 - q_L)^m \to \mathbb{1}\{q_L = 0\}$ pointwise, and every term below is bounded by 1. Dominated convergence gives, for $b \in \{L, H\}$,

$$\begin{aligned} \mathbb{E}\big[q_b\,(1-q_L)^m (1-q_H)^n\big] &\longrightarrow \mathbb{E}\big[q_b\,(1-q_H)^n \mathbb{1}\{q_L = 0\}\big], \\ R_{m,n} = \mathbb{E}\big[(1-q_L)^m (1-q_H)^n\big] &\longrightarrow \mathbb{E}\big[(1-q_H)^n \mathbb{1}\{q_L = 0\}\big] > 0. \end{aligned} \tag{13}$$

For $b = L$ the first limit vanishes since $q_L \mathbb{1}\{q_L = 0\} = 0$; dividing the limits gives both claims. □

A mixture can win through the mean term alone. A zero covariance term does not make mixing useless. Take two equally likely types of problems with $(q_L, q_H) = (0.2, 0.5)$ and $(0.4, 0.7)$, costs $c_L = 1$ and $c_H = 2$, no switch charge, and a budget $C = 3$, so that $(3, 0)$ and $(0, 1)$ are the largest pure pairs and $(1, 1)$ the largest mixed pair. Here $\mu_L = 0.3$, $\mu_H = 0.6$ and $\operatorname{Var}(q_L) = \operatorname{Cov}(q_L, q_H) = 0.01$, so the covariance term is exactly zero. Table 5 evaluates every schedule with equation 1: the mixture $(1, 1)$ solves 0.710 against 0.636 for the best of the pure schedules, and Proposition 2 assigns all of $S(1,1) - S(2,0) = 0.21$ to the mean term $(1 - 0.3)(0.6 - 0.3) = 0.21$.

Table 5: Two types of problems with a zero covariance term. Equal weights, $c_L = 1$, $c_H = 2$, $C = 3$. Solve probability of each schedule (m, n) on each type and on average over both; the last column checks Proposition 2.

Type	(q_L, q_H)	$S(3,0)$	$S(0,1)$	$S(1,1)$	$S(2,0)$	$S(1,1) - S(2,0)$
A	$(0.2, 0.5)$	0.488	0.500	0.600	0.360	0.240
B	$(0.4, 0.7)$	0.784	0.700	0.820	0.640	0.180
Average		0.636	0.600	0.710	0.500	$0.210 = 0.21 + 0.00$

B.3 Proof of Proposition 4

Proof of Proposition 4. A policy stops at the first accepted answer, so before it stops it has observed only rejections. Its choice for attempt t is therefore a function of the history "$t-1$ rejections", which is the same on every problem that reaches attempt t, and the policy runs one fixed sequence σ on all problems. By Proposition 1, $S(\sigma) = S(m,n)$ for the counts (m,n) of σ. The scheduled cost of σ, which charges every attempt and every change of precision whether or not it is reached, is

$$\mathrm{cost}_{\mathrm{sched}}(\sigma) = m\,c_L + n\,c_H + c_{\mathrm{sw}} \sum_{t=2}^{m+n} \mathbb{1}[\sigma_t \neq \sigma_{t-1}] \;\geq\; m\,c_L + n\,c_H + c_{\mathrm{sw}}\,\mathbb{1}[m>0, n>0], \quad (14)$$

with equality exactly for the blocked sequences $L^m H^n$ and $H^n L^m$, which change precision at most once. Every policy whose scheduled cost fits C therefore has $(m,n) \in \mathcal{F}_C$, and every pair in $\mathcal{F}_C$ is reached by a blocked sequence, so the largest solve rate over all such policies is $\max_{(m,n)\in\mathcal{F}_C} S(m,n)$. A randomized policy draws one such sequence per problem, so its solve rate is an average of the values $S(m,n)$ with $(m,n) \in \mathcal{F}_C$ and cannot exceed this maximum either. □

The argument needs a policy that sees nothing but acceptance and rejection. A policy that reads the content of a failed answer, as self-repair does, or features of the problem, as a router does, can choose differently on different problems of the pool; the first breaks (A1), and the second is one of the comparators of Table 1. Under an expected-cost budget instead of a scheduled one, the solve rate still depends on the pair (m,n) alone, and the order enters only through Lemma 1 and Proposition 6.

B.4 Proof of Theorem 1

Throughout, $n=0$, $R_r = \mathbb{E}[(1-q_L)^r]$ and $\mathbb{E}_r[f] = \mathbb{E}[f\,(1-q_L)^r]/R_r$ is the expectation over the open problems. The assumption $\mathrm{P}(0<q_L<1)>0$ keeps R_r, B_r and $1-h_r^L$ positive for every r, so all hazards below are well defined. The proof rests on one identity for a single failure, the covariance form of a reweighting that population genetics knows as the Price equation (Price, 1970).

Lemma 2 (One more failure)**.** *For every f with $\mathbb{E}_r|f| < \infty$,*

$$\mathbb{E}_{r+1}[f] - \mathbb{E}_r[f] = -\frac{\mathrm{Cov}_r(f, q_L)}{1-h_r^L}. \quad (15)$$

Proof. Divide the numerator and the denominator of $\mathbb{E}_{r+1}[f]$ by R_r, and use $\mathbb{E}_r[f q_L] = \mathbb{E}_r[f]\,\mathbb{E}_r[q_L] + \mathrm{Cov}_r(f, q_L)$ with $\mathbb{E}_r[q_L] = h_r^L$:

$$\mathbb{E}_{r+1}[f] = \frac{\mathbb{E}\big[f\,(1-q_L)^{r+1}\big]}{\mathbb{E}\big[(1-q_L)^{r+1}\big]} = \frac{\mathbb{E}_r\big[f\,(1-q_L)\big]}{\mathbb{E}_r\big[1-q_L\big]} = \frac{\mathbb{E}_r[f] - \mathbb{E}_r[f q_L]}{1-h_r^L}$$
$$= \frac{\mathbb{E}_r[f] - \mathbb{E}_r[f]\,h_r^L - \mathrm{Cov}_r(f,q_L)}{1-h_r^L} = \mathbb{E}_r[f] - \frac{\mathrm{Cov}_r(f,q_L)}{1-h_r^L}. \qquad \square$$

Proof of Theorem 1. *(i)* Take $f = q_L$ in equation 15 of Lemma 2: $\mathrm{Cov}_r(q_L, q_L) = \mathrm{Var}_r(q_L)$, so $h_{r+1}^L = h_r^L - \mathrm{Var}_r(q_L)/(1-h_r^L) \leq h_r^L$, with equality only when all open problems share one value of q_L. With $f = q_H$ instead, $h_{r+1}^H = h_r^H - \mathrm{Cov}_r(q_L, q_H)/(1-h_r^L)$, which falls when $\mathrm{Cov}_r(q_L, q_H) > 0$, that is, when the open problems that are easy for L tend to be easy for H as well.

(ii) The factor R_r cancels in the ratio of the two hazards, so $\rho_r = A_r/B_r$ with $A_r = \mathbb{E}[q_H(1-q_L)^r]$ and $B_r = \mathbb{E}[q_L(1-q_L)^r]$. On the event $\{q_L = 0\}$ the weight $(1-q_L)^r$ equals 1, and on $\{q_L > 0\}$ we may write $q_H = (q_H/q_L)\,q_L$. Hence

$$\rho_r = \frac{h_r^H}{h_r^L} = \frac{A_r}{B_r} = \frac{\mathbb{E}\big[q_H \mathbb{1}\{q_L=0\}\big] + \mathbb{E}\big[q_H(1-q_L)^r \mathbb{1}\{q_L>0\}\big]}{B_r}$$
$$= \frac{a_0}{B_r} + \frac{\mathbb{E}\big[(q_H/q_L)\,q_L(1-q_L)^r \mathbb{1}\{q_L>0\}\big]}{\mathbb{E}\big[q_L(1-q_L)^r\big]}$$
$$= \frac{a_0}{B_r} + \mathbb{E}_{\nu_r}\big[g\big], \qquad g = q_H/q_L. \quad (16)$$

The law ν_{r+1} is ν_r reweighted by $1 - q_L$, exactly as $\mathbb{E}_{r+1}$ is $\mathbb{E}_r$ reweighted by $1 - q_L$, so Lemma 2 holds with ν_r in place of $\mathbb{E}_r$ and $\mathbb{E}_{\nu_r}[q_L]$ in place of h_r^L. The integrability condition holds because $\mathbb{E}_{\nu_r}[g] \le 1/B_r$. Therefore

$$\rho_{r+1} - \rho_r = a_0\Big(\frac{1}{B_{r+1}} - \frac{1}{B_r}\Big) - \frac{\mathrm{Cov}_{\nu_r}(g, q_L)}{1 - \mathbb{E}_{\nu_r}[q_L]}, \tag{17}$$

which is equation 4, the drift of Theorem 1. Since $(1 - q_L)^{r+1} \le (1 - q_L)^r$, the sequence B_r is non-increasing and the first term is never negative, while the second has the sign of $-\mathrm{Cov}_{\nu_r}(g, q_L)$.

(iii) If $\mathrm{Cov}_{\nu_r}(g, q_L) \le 0$ for every r, both terms of equation 4 are non-negative and ρ_r is non-decreasing. The reweighting that defines ν_r depends on q_L alone, so the conditional law of q_H given q_L is the same under ν_r and under $\mathcal{D}$, and by the law of total covariance

$$\mathrm{Cov}_{\nu_r}(g, q_L) = \mathrm{Cov}_{\nu_r}\big(\bar{g}(q_L),\, q_L\big), \qquad \bar{g}(t) = \frac{\mathbb{E}[q_H \mid q_L = t]}{t}. \tag{18}$$

A non-increasing function of a random variable has non-positive covariance with that variable (Esary et al., 1967), so the condition holds for every r when $\bar{g}$ is non-increasing. Now let $\tau = (c_H + c_{\mathrm{sw}})/c_L$. The comparison $h_r^H/(c_H + c_{\mathrm{sw}}) > h_r^L/c_L$ is the same as $\rho_r > \tau$, and the r at which a non-decreasing sequence exceeds a fixed level form a set $\{r \ge r_0\}$. Finally, let $a_0 > 0$. The integrand of $B_r = \mathbb{E}[q_L(1 - q_L)^r]$ tends to zero pointwise and is bounded by 1, so $B_r \to 0$ by dominated convergence, and $\rho_r \ge a_0/B_r \to \infty$ whether or not the condition holds. Hence ρ_r exceeds the level τ for all large r, and the set $\{r \ge r_0\}$ is not empty, so after enough failures of L the switch wins. □

The reweighting from ν_r to ν_{r+1} has density proportional to $1 - q_L$, a decreasing function of q_L, so each failure moves ν_r toward small q_L, that is, down in the likelihood-ratio order (Shaked & Shanthikumar, 2007); part (iii) is the consequence of this move for the ratio q_H/q_L and hence for ρ_r.

Checks on the measured pools. At the reference setting the ratio of the two yields per second, ρ_r/τ, rises at every step from 0.65 at $r = 0$ to 1.93 at $r = 8$ and passes one between $r = 1$ and $r = 2$ (Table 18); in the median context it rises from 0.52 to 1.72 over $r = 0, \ldots, 12$ and passes one at $r = 3$ (Fig. 9a). Over all 288 contexts, the share in which the switch wins equals the share that have crossed at least once, at every r up to 12 (Fig. 9), so no context crosses back within twelve failures.

B.5 Proof of Proposition 5 and Concentration of the Calibration

Let $\hat{p}_{b,i} = s_{b,i}/k$ be the observed success rate of b on calibration problem x_i. For the split-half estimate, k is even and the k attempts of L on x_i are split into two disjoint halves with success rates $\hat{p}_{L,i}^A$ and $\hat{p}_{L,i}^B$. The estimates of $\mathrm{Var}(q_L)$ and $\mathrm{Cov}(q_L, q_H)$ are

$$\begin{aligned}
\widehat{\mathrm{Var}}(q_L) &= \frac{1}{N}\sum_{i=1}^{N} \hat{p}_{L,i}^A\, \hat{p}_{L,i}^B - \frac{1}{N(N-1)}\sum_{i \ne i'} \hat{p}_{L,i}\, \hat{p}_{L,i'}, \\
\widehat{\mathrm{Cov}}(q_L, q_H) &= \frac{1}{N}\sum_{i=1}^{N} \hat{p}_{L,i}\, \hat{p}_{H,i} - \frac{1}{N(N-1)}\sum_{i \ne i'} \hat{p}_{L,i}\, \hat{p}_{H,i'}.
\end{aligned} \tag{19}$$

The second sum of each is a U-statistic of order two (Hoeffding, 1948). The diagnostic variant computes the two estimates and the mean shift of the observed rates from $k = 8$ attempts, and decides from them whether to switch and after how many attempts, without enumerating pairs (Appendix D).

Lemma 3 (Unbiased estimate of a survival probability)**.** *If $s \sim \mathrm{Binomial}(k, q)$ and $m \le k$, then $\mathbb{E}\big[\binom{k-s}{m}/\binom{k}{m}\big] = (1 - q)^m$, so for every $q \in [0, 1]$ the ratio is an unbiased estimate of $(1 - q)^m$.*

Proof. The ratio is zero when $s > k - m$. Using $\binom{k}{s}\binom{k-s}{m} = \binom{k}{m}\binom{k-m}{s}$, since both sides count the ways to choose disjoint sets of s and m items out of k, and then the binomial theorem,

$$\begin{aligned}\mathbb{E}\Big[\binom{k-s}{m}\Big/\binom{k}{m}\Big] &= \sum_{s=0}^{k-m}\binom{k}{s}q^s(1-q)^{k-s}\,\frac{\binom{k-s}{m}}{\binom{k}{m}} = \sum_{s=0}^{k-m}\binom{k-m}{s}q^s(1-q)^{k-s}\\ &= (1-q)^m\sum_{s=0}^{k-m}\binom{k-m}{s}q^s(1-q)^{k-m-s}\\ &= (1-q)^m\big(q+1-q\big)^{k-m} = (1-q)^m.\end{aligned}$$

□

Proof of Proposition 5. Given x_i, the counts $s_{L,i}$ and $s_{H,i}$ are independent binomials with success probabilities $q_L(x_i)$ and $q_H(x_i)$, because attempts of the two configurations are independent under (A1). Lemma 3 and independence give $\mathbb{E}[\hat{F}_{L,i}(m)\hat{F}_{H,i}(n) \mid x_i] = (1-q_L(x_i))^m(1-q_H(x_i))^n$, and averaging over the calibration problems, which are drawn from $\mathcal{D}$, gives $\mathbb{E}[\hat{S}(m,n)] = S(m,n)$. For the split-half estimate, the two halves of problem x_i are independent given x_i, and distinct problems are independent, so

$$\begin{aligned}\mathbb{E}\big[\widehat{\mathrm{Var}}(q_L)\big] &= \frac{1}{N}\sum_i \mathbb{E}\Big[\mathbb{E}\big[\hat{p}^A_{L,i} \mid x_i\big]\,\mathbb{E}\big[\hat{p}^B_{L,i} \mid x_i\big]\Big] - \frac{1}{N(N-1)}\sum_{i\neq i'}\mathbb{E}\big[\hat{p}_{L,i}\big]\,\mathbb{E}\big[\hat{p}_{L,i'}\big]\\ &= \mathbb{E}\big[q_L^2\big] - \mu_L^2 = \mathrm{Var}(q_L),\end{aligned} \tag{20}$$

and in the same way $\mathbb{E}[\widehat{\mathrm{Cov}}(q_L, q_H)] = \mathbb{E}[q_L q_H] - \mu_L\mu_H = \mathrm{Cov}(q_L, q_H)$; no split is needed here because the two configurations never share an attempt. The plug-in estimate is the sample variance of the $\hat{p}_{L,i}$, which is unbiased for $\mathrm{Var}(\hat{p}_{L,i})$, and by the law of total variance

$$\mathrm{Var}(\hat{p}_{L,i}) = \mathrm{Var}\left(\mathbb{E}[\hat{p}_{L,i} \mid x_i]\right) + \mathbb{E}\big[\,\mathrm{Var}(\hat{p}_{L,i} \mid x_i)\big] = \mathrm{Var}(q_L) + \mathbb{E}\Big[\frac{q_L(1-q_L)}{k}\Big].$$

□

At the reference setting $\mu_L = 0.514$ and $\mathrm{Var}(q_L) = 0.1573$, so $\mathbb{E}[q_L(1-q_L)] = \mu_L - \mu_L^2 - \mathrm{Var}(q_L) = 0.0925$. With only $k = 8$ the plug-in variance therefore overstates $\mathrm{Var}(q_L)$ by 0.012, more than half of the covariance term 0.0217 whose estimate it enters. The simulation with known truth in Table 13 shows a plug-in bias of 0.020 to 0.027 at $k = 8$ on each of its three constructions.

Proposition 7 (Concentration of the calibration)**.** *Under the assumptions of Proposition 5, suppose every pair in $\mathcal{F}_C$ has $m, n \le k$. For every $\delta \in (0,1)$, with probability at least $1-\delta$ over the calibration problems,*

$$\max_{(m,n)\in\mathcal{F}_C}\big|\hat{S}(m,n) - S(m,n)\big| \le \varepsilon_N = \sqrt{\frac{\log(2|\mathcal{F}_C|/\delta)}{2N}}, \tag{21}$$

and hence the pair chosen by equation 3 satisfies $S(m^\star, n^\star) \ge \max_{(m,n)\in\mathcal{F}_C} S(m,n) - 2\varepsilon_N$.

Proof. For a fixed pair, $\hat{S}(m,n)$ is one minus the mean of N independent terms $\hat{F}_{L,i}(m)\hat{F}_{H,i}(n) \in [0,1]$ with mean $1 - S(m,n)$, so Hoeffding's inequality (Hoeffding, 1963) gives $\mathrm{P}(|\hat{S} - S| > \varepsilon) \le 2e^{-2N\varepsilon^2}$, and a union bound over the $|\mathcal{F}_C|$ pairs gives the first claim of the proposition. On that event, with (m°, n°) a best pair, $S(m^\star, n^\star) \ge \hat{S}(m^\star, n^\star) - \varepsilon_N \ge \hat{S}(m^\circ, n^\circ) - \varepsilon_N \ge S(m^\circ, n^\circ) - 2\varepsilon_N$. □

The bound treats every pair on its own. The pairs that compete for the maximum share most of their attempts, so their estimated difference is far less noisy than each estimate, and the enumeration settles faster than the bound suggests: 128 calibration problems come within 0.04 points of 512 (Table 2).

B.6 Proof of Proposition 3 and a Decomposition of the Gain

Proof of Proposition 3. For every problem x and every feasible pair, $1-(1-q_L(x))^m(1-q_H(x))^n \le S^\star_{\mathrm{mix}}(x)$ by the definition of $S^\star_{\mathrm{mix}}(x)$ as the largest solve probability over the feasible pairs. Taking expectations,

$$S(m,n) \le \mathbb{E}[S^\star_{\mathrm{mix}}] = \mathbb{E}[S^\star_{\mathrm{pure}}] + \Delta = S_{\mathrm{pure}} + V + \Delta. \tag{22}$$

The bound holds for every population pair, calibrated or not, since the pair enters only on the left. □

Let $G = S(m^\star, n^\star) - S_{\text{pure}}$ be the gain of the calibrated schedule over the better pure schedule. Rearranging the same terms gives an exact account of it:

$$G = \underbrace{V}_{\text{perfect information}} + \underbrace{\Delta}_{\text{value of mixing}} - \underbrace{\big(\mathbb{E}[S^\star_{\text{mix}}] - S(m^\star, n^\star)\big)}_{\text{cost of not seeing } q_b(x) \geq 0}. \tag{23}$$

On the twelve wall-clock settings of Table 1, the means are $V = 6.06$, $\Delta = 0.67$ and $G = 2.49$ points (Table 8), so the last term is about 4.2 points on average: a schedule that sees nothing but its own failures recovers about 41% of what perfect information about the better precision would give.

B.7 Prices Under the Memory-Time Budget

Under the memory-time budget every second of an attempt is charged with the memory held. With only L resident an attempt of L costs $c_L M_L$, with only H resident an attempt of H costs $c_H M_H$, and with both resident every attempt, of either precision, is charged $M_L + M_H$ for each of its seconds:

$$\text{memory-time}(m, n) = \begin{cases} m\, c_L\, M_L, & n = 0, \\ n\, c_H\, M_H, & m = 0, \\ \big(m c_L + n c_H + c_{\text{sw}}\big)\,(M_L + M_H), & m, n > 0. \end{cases} \tag{24}$$

The budget of a setting is $C_{\text{m}} = C_{\text{w}} \times M_H$, which gives fixed H exactly its wall-clock allowance and every schedule that holds less memory more seconds, in inverse proportion to the memory held.

Proposition 8 (Attempts under the memory-time budget). *Let $\kappa = (M_L + M_H)/M_L$ and $\bar{m} = \lfloor C_{\text{m}}/(c_L M_L) \rfloor$, the number of attempts of L alone that fit. Every feasible mixed pair (m, n) with $m, n \geq 1$ satisfies*

$$m + n\,\frac{c_H}{c_L} + \frac{c_{\text{sw}}}{c_L} \leq \frac{C_{\text{m}}}{c_L(M_L + M_H)} < \frac{\bar{m} + 1}{\kappa}. \tag{25}$$

Proof. Divide the feasibility condition $(mc_L + nc_H + c_{\text{sw}})(M_L + M_H) \leq C_{\text{m}}$ by $c_L(M_L + M_H)$, and use $C_{\text{m}}/(c_L M_L) < \bar{m} + 1$, which holds by the definition of $\bar{m}$, together with $M_L + M_H = \kappa M_L$. □

At the reference setting $\kappa = 86.5/19.1 = 4.53$ and $\bar{m} = 19$, so a mixed pair must satisfy $m + 1.64\, n \leq 4.26$, and the largest one is $(2, 1)$, which uses 1382 GB·s of the 1617.6 available. One attempt of H with both checkpoints resident costs as much memory-time as 7.4 attempts of L alone (618.4 against 83.5 GB·s, Table 9). The survivors show why this price is not paid: after two failures 94.5 of the 240 problems are open, the next six attempts of L solve 41.3 of them, and one attempt of H would solve about $94.5 \times 0.237 = 22.4$ (Table 18). The enumeration therefore returns $(19, 0)$, which solves 86.78 points against 69.97 for $(2, 1)$ (Table 22), and it keeps only L resident, at 19.1 GB.

The budget scales C_{w} by M_H, so fixed H keeps the three attempts it has under wall clock. With M_L in its place the budget would be $24.0 \times 19.1 = 458$ GB·s: fixed L would afford five attempts, as under wall clock, and not even a single attempt of H with both checkpoints resident would fit. With either multiplier, a second of an attempt with L alone resident costs $1/\kappa$ of a second with both checkpoints resident, and this ratio, $1/4.53$, decides whether the high-precision block is ever bought.

C The Registered Criterion

This appendix states which hypothesis the registered criterion of Section 3.3 tests, shows with two types of problems that a switch can gain a lot even when the criterion fails, and gives the numbers behind the criterion and behind the value of perfect information on the wall-clock settings of Table 1.

What the criterion tests. Three solve rates are in play for a fixed setting and budget. The population pure rate S_{pure} is what repeating the better precision achieves, one schedule for every problem. The per-problem pure rate $\mathbb{E}[S^\star_{\text{pure}}]$ is what an oracle achieves that picks, for every problem separately, the better of the two pure schedules. The per-problem mixed rate $\mathbb{E}[S^\star_{\text{mix}}]$ lets the same oracle pick

any feasible pair. They are ordered as $S_{\text{pure}} \le \mathbb{E}[S^\star_{\text{pure}}] \le \mathbb{E}[S^\star_{\text{mix}}]$, and the two gaps are V and Δ. The criterion tests Δ alone: it asks whether mixing adds anything once the better precision has already been chosen for each problem. It says nothing about V, and Proposition 3 allows a population schedule to collect part of V while $\Delta = 0$, as in the example with two types of problems that follows.

Why the value of mixing is small. Fix a problem and write $\alpha = -\log(1 - q_L(x))$ and $\beta = -\log(1 - q_H(x))$. A pair (m, n) fails on it with probability $e^{-(m\alpha+n\beta)}$, so the best pair on this problem maximizes $m\alpha + n\beta$ over the feasible set. If attempts could be split, a linear objective over the triangle $mc_L + nc_H \le C$ would be largest at a vertex, so the whole budget would go to the precision with the larger α/c_L or β/c_H, and the switch charge further raises the cost of a mixed pair. A mixed pair can win only because attempts are whole: a pure schedule may leave part of the budget unused, and a mixed pair may spend the rest. This happens only when β/α falls in a narrow band set by c_L, c_H and C, so Δ is small unless many problems fall in that band and is zero when none does.

A switch can gain a lot while the criterion fails. Let both attempts cost one unit, let the budget allow two attempts, and let half of the problems have $(q_L, q_H) = (0.9, 0.1)$ and the other half $(0.1, 0.9)$. Table 6 evaluates every schedule with equation 1. The fixed schedules LL and HH each solve 0.99 on one type and 0.19 on the other, 0.59 on average. The mixed schedule LH solves $1 - 0.1 \times 0.9 = 0.91$ on both types, 32 points more. Yet $\Delta = 0$, because the per-problem oracle already repeats the right precision twice on every problem and solves 0.99, which no mixture can beat. The switch collects 0.32 of the value of perfect information $V = 0.40$ without knowing which type a problem is, because its first attempt fails mostly on the type where the other precision is right.

Table 6: Two types of problems on which the criterion fails and a switch gains 32 points. Equal attempt costs, a budget of two attempts, no switch charge, equal weights. Solve probability of every schedule on each type and on average, and of the two per-problem oracles; the pure oracle runs LL on type A and HH on type B.

Type	(q_L, q_H)	LL	HH	LH	per-problem pure	per-problem mixed
A	$(0.9, 0.1)$	0.99	0.19	0.91	0.99	0.99
B	$(0.1, 0.9)$	0.19	0.99	0.91	0.99	0.99
Average		0.59	0.59	0.91	0.99	0.99

$$S_{\text{pure}} = 0.59, \quad \mathbb{E}[S^\star_{\text{pure}}] = 0.99, \quad \mathbb{E}[S^\star_{\text{mix}}] = 0.99, \quad V = 0.40, \quad \Delta = 0,$$
$$G = 0.91 - 0.59 = 0.32 \le V + \Delta.$$

The registered settings. Table 7 lists the twelve settings of the criterion with Δ and its 95% bootstrap interval. Three settings meet the registered threshold, Qwen2.5-7B with GPTQ-W4 on code and on math and Qwen2.5-72B with GPTQ-W4 on math, and four settings have $\Delta = 0$ exactly, since no problem there has a mixed pair as its best. The criterion required eight such settings and is not met.

Table 7: The twelve registered settings. Value of mixing Δ in points with its 95% bootstrap interval. The registered criterion asked for $\Delta \ge 2.0$ with a positive lower end in at least 8 of the 12 settings; the last column marks the 3 settings that meet it, all three with GPTQ-W4, and four of the twelve settings have $\Delta = 0$ exactly.

Model	Recipe	Family	Δ	low	high	met
Qwen2.5-7B	GPTQ-W4	Code	2.34	1.96	2.72	✓
Qwen2.5-7B	GPTQ-W4	Math	2.12	1.71	2.53	✓
Qwen2.5-7B	AWQ-W4	Code	0.00	0.00	0.00	
Qwen2.5-7B	AWQ-W4	Math	0.68	0.49	0.86	
Qwen2.5-32B	GPTQ-W4	Code	0.00	0.00	0.00	
Qwen2.5-32B	GPTQ-W4	Math	0.00	0.00	0.00	
Qwen2.5-32B	AWQ-W4	Code	0.00	0.00	0.01	
Qwen2.5-32B	AWQ-W4	Math	0.39	0.25	0.53	
Qwen2.5-72B	GPTQ-W4	Code	0.45	0.32	0.57	
Qwen2.5-72B	GPTQ-W4	Math	2.28	1.80	2.76	✓
Qwen2.5-72B	AWQ-W4	Code	0.00	0.00	0.00	
Qwen2.5-72B	AWQ-W4	Math	0.04	0.01	0.07	

The same arithmetic on the settings of the main table. The definitions of Section 3.3 apply in the same way to the twelve wall-clock settings of Table 1 (Table 8). The value of mixing Δ is at most 2.34 points and below 0.5 points in nine settings, while V lies between 4.48 and 7.93 points, and RetryPrec collects 8% to 57% of it in each setting and 2.49 of 6.06 points on average over the twelve.

The largest shares are on reasoning and agent, where Δ stays below half a point. On these pools the switch pays through the choice of precision that failures make possible, as in the example of Table 6.

Table 8: Value of perfect information, value of mixing and the gain of RetryPrec on the twelve wall-clock settings of Table 1, in points. S_{pure} is the better of fixed L and fixed H; $V = \mathbb{E}[S^{\star}_{\text{pure}}] - S_{\text{pure}}$ and $\Delta = \mathbb{E}[S^{\star}_{\text{mix}} - S^{\star}_{\text{pure}}]$ are read from the two oracle rows; G is RetryPrec minus S_{pure}. The bound $G \le V + \Delta$ of Proposition 3 holds in every row, and the mean row gives G/V as the mean of G divided by the mean of V.

Family	Model	Fixed L	Fixed H	$\mathbb{E}[S^{\star}_{\text{pure}}]$	$\mathbb{E}[S^{\star}_{\text{mix}}]$	V	Δ	G	G/V
Code	7B	74.78	67.72	79.26	81.60	4.48	2.34	1.08	24%
Code	32B	72.12	71.32	80.05	80.05	7.93	0.00	3.43	43%
Code	72B	78.13	74.09	83.53	83.98	5.40	0.45	2.21	41%
Math	7B	76.16	70.67	82.55	84.67	6.39	2.12	2.39	37%
Math	32B	76.30	72.46	81.78	81.78	5.48	0.00	0.42	8%
Math	72B	73.05	65.15	77.74	80.02	4.69	2.28	2.22	47%
Reasoning	7B	75.26	77.66	85.29	85.30	7.63	0.01	3.39	44%
Reasoning	32B	78.71	75.56	84.73	85.00	6.02	0.27	3.16	52%
Reasoning	72B	76.81	73.26	83.15	83.56	6.34	0.41	3.44	54%
Agent	7B	77.07	76.11	84.25	84.26	7.18	0.01	4.11	57%
Agent	32B	78.74	80.41	86.79	86.81	6.38	0.02	3.28	51%
Agent	72B	74.32	64.10	79.11	79.18	4.79	0.07	0.75	16%
Mean		75.95	72.38	82.35	83.02	6.06	0.67	2.49	41%

Under the memory-time budget. The two oracles are equal on average under memory-time, 90.26 points on the twelve settings, so Δ is zero on average there, and G is zero because RetryPrec equals fixed L. The value of perfect information under that budget is only $90.26 - 88.81 = 1.45$ points, which is small because the budget already buys many attempts of L for every problem in the pool.

D Method and Experimental Details

D.1 Diagnostic Variant

Algorithm 1 in Section 4.1 states RetryPrec as run in every experiment: the calibration scores every feasible pair with the unbiased estimate of equation 3 and keeps the best one, with ties going to the smaller of the high-precision blocks. The diagnostic variant replaces the enumeration by the two-attempt diagnostic of Proposition 2 and a choice between two switch points. From $k = 8$ attempts per configuration it computes the estimates of equation 19 and the mean shift between the observed rates, and it decides from these numbers whether to switch. When it switches, it considers only low-precision blocks of 2 and of 6 attempts, each followed by the high-precision block that fills the rest of the budget, and keeps the one with the larger estimate $\hat{S}$ of equation 3, computed from the same eight attempts. It scores two pairs instead of every feasible one and needs one eighth of the calibration attempts; at Qwen2.5-32B it costs 0.49 points on average over the four families (Table 2).

D.2 Pools, Tasks and Models

For every model and task family we build two disjoint sets of problems, which all eight recipes share. The pool has 240 evaluation problems, and on each of them we run 64 attempts of L and 64 of H at temperature 0.7; the fraction of accepted attempts estimates $q_b(x)$. The calibration set has 512 further problems with 64 attempts per configuration, and RetryPrec sees only this set when it calibrates. The four families follow the format of LiveCodeBench for code, AIME for math, GPQA for reasoning, and agent tasks whose results are checked automatically by a program. Every strategy is evaluated on the same pool, so all comparisons between strategies are paired by problem. The nine models, Qwen2.5-7B, 14B, 32B and 72B, Llama-3.1-8B and 70B, DeepSeek-R1-Distill-Qwen-7B and 32B, and Mistral-Small-24B, are each served as a BF16 checkpoint H and as eight quantized copies L: GPTQ-W4, AWQ-W4, SmoothQuant-W8A8, FP8, QAT-W4, QuaRot, SpinQuant and AQLM/QuIP#. The 288 contexts of Fig. 3c are the combinations of the nine models, eight recipes and four families.

D.3 Cost Accounting and Budgets

Unit costs are measured on our serving stack for each configuration. An attempt is charged its full time, including the prefill of its own prompt, which is 3.0% to 10.3% of an attempt across the nine

models (Table 10); the KV cache is never reused across precisions, so a switch adds only its own charge c_{sw}, which is 56 to 206 ms over the models and 94 ms at Qwen2.5-32B (Table 25). The scheduled cost of a sequence is the sum of its attempt costs plus c_{sw} for every change of precision: one change for a blocked pair, three for the alternation $LHLH$ and four for $LHLHLL$. The expected cost charges each attempt and each change only when it is reached (Lemma 1). A pair is feasible when its scheduled cost fits the budget. Resident memory is the recorded peak of each checkpoint, 19.1 GB for L and 67.4 GB for H at the reference setting, and 86.5 GB when both stay resident. The wall-clock budget is set per family, and the memory-time budget is $C_{\mathrm{m}} = C_{\mathrm{w}} \times M_H$, priced by equation 24. Table 9 lists the resulting prices and the attempts that each budget buys at Qwen2.5-32B.

Table 9: Unit prices under the two budgets at Qwen2.5-32B with GPTQ-W4, per family. Attempt costs c_L and c_H (s), the wall-clock budget C_{w} (s) and the memory-time budget $C_{\mathrm{m}} = C_{\mathrm{w}} \times 67.4$ GB (GB·s); the memory-time price of one attempt with L alone ($c_L \times 19.1$) and with both resident ($c_L \times 86.5$, $c_H \times 86.5$); the largest number of attempts that each budget buys; and the $m^\star$ that the enumeration returns under memory-time.

					Price of one attempt (GB·s)			Attempts the budget buys				
Family	c_L	c_H	C_{w}	C_{m}	L alone	L, both	H, both	L, wall	L, mem.	L with H	H with L	$m^\star$
Code	4.37	7.15	24.0	1617.6	83.5	377.9	618.4	5	19	4	2	19
Math	4.67	9.39	28.8	1941.1	89.3	404.1	811.9	6	21	4	2	21
Reasoning	4.07	6.38	30.0	2022.0	77.9	352.4	551.8	7	25	5	3	25
Agent	7.12	12.41	54.6	3680.0	136.1	616.0	1073.4	7	27	5	3	27

D.4 STRATEGIES

All 21 strategies of Table 1 are run and evaluated on the same pool and charged by the same unit costs; a strategy that needs a router or a reward model is also charged for it. For a strategy with two configurations at the same precision, m and n count the attempts of the first and of the second one.

References. *BF16, one attempt* runs H once. *Oracle, pure* and *oracle, mixture* choose, for every problem separately and from its true $q_L(x)$ and $q_H(x)$, the best feasible pure schedule and the best feasible pair; no deployment can run them, and they define $\mathbb{E}[S^\star_{\mathrm{pure}}]$ and $\mathbb{E}[S^\star_{\mathrm{mix}}]$ used in Section 3.3.

Fixed schedules and diversity at one precision. *Fixed* L and *fixed* H repeat one precision as often as the budget allows, which is the common practice. *More samples* draws further samples of L and *best-of-*n *with an outcome reward model* selects among samples of L; with a perfect verifier and a stop at the first accepted answer, both run the attempts of fixed L and coincide with it in solve rate. *Temperature* T=1.0 repeats L at a higher temperature. $L{+}L'$ allocates the attempts between two 4-bit recipes of the same model, GPTQ-W4 and AWQ-W4, and L + *temperature* between L at T=0.7 and at T=1.0, both by the enumeration of equation 3; these stay at low precision, and L + temperature needs a single resident checkpoint, as fixed L, more samples and the temperature row do.

Allocation between the two precisions. *Random pair* draws one feasible pair (m, n) per family and runs it, as blocks in Table 1 and interleaved in Table 21. *Fixed* m spends about half of the budget on L, $m = \lfloor C/2c_L \rfloor$ at matched wall clock, and the rest on H. *Transferred* m takes the switch point from another setting instead of calibrating it on its own problems. *Cross-configuration allocation* splits the attempts among configurations of the same bit width in the manner of Zhang et al. (2025); on our pools it chooses the attempts of $L{+}L'$ and thus coincides with it in each of the twelve settings.

Cascades and routing. The *FrugalGPT cascade* calls L first and escalates to H on rejection as in Chen et al. (2024b). *Hybrid LLM routing* and *RouteLLM* route each problem to one of the two precisions up front (Ding et al., 2024; Ong et al., 2025); their pairs are fractional means over problems.

Decoding and selection. *Speculative decoding* runs H with a draft model and reproduces the output distribution of H, and *best-of-*n *with a process reward model* scores samples of H (Lightman et al., 2024); on our pools both coincide with fixed H in solve rate, since both run the attempts of fixed H.

Ours. *RetryPrec* is Algorithm 1 with $N = 512$, $k = 64$, and *RetryPrec, diagnostic* the variant above.

The cascade and the diagnostic variant both run a block of L followed by a block of H, and on eleven of the twelve wall-clock settings of Table 1, all but Qwen2.5-72B math, they choose the same pair.

D.5 Validity Checks

Table 10 records, for each model, the checks on which the accounting rests. The blocked schedule and the same pre-ordered sequence $L^m H^n$ run on a shared random stream agree in solve rate; the visible-test verifier accepts wrong answers at a rate of 0.0067 to 0.0079 across the nine models; the prefill is charged inside every attempt; and the KV cache is never reused across the two precisions.

Table 10: Validity checks per model. Difference in solve rate between the blocked schedule and the same pre-ordered sequence $L^m H^n$ run on a shared random stream; false-accept rate of the visible-test verifier; prefill share of the time of an attempt, which is charged inside every attempt; and KV-cache reuse across precisions.

Model	blocked vs. pre-ordered	false accept	prefill	KV reuse
Qwen2.5-7B	0	0.0068	0.0335	0
Qwen2.5-14B	0	0.0071	0.0482	0
Qwen2.5-32B	0	0.0076	0.0634	0
Qwen2.5-72B	0	0.0069	0.0298	0
Llama-3.1-8B	0	0.0078	0.1027	0
Llama-3.1-70B	0	0.0072	0.0322	0
R1-Distill-Qwen-7B	0	0.0075	0.0409	0
R1-Distill-Qwen-32B	0	0.0067	0.0564	0
Mistral-Small-24B	0	0.0079	0.0308	0

D.6 Cost of the Calibration

The main table does not charge the calibration. Table 11 compares ways of spending it at the reference setting, and Table 12 gives its cost per family. Estimating a pair for every served problem from 128 attempts of its own would raise the solve rate on code from 75.55 to 79.45 points but costs 737 s per problem, about 32 times the deployment cost, so it serves only as a reference. One calibration per family is enough in practice: with 512 problems it costs 3.4×10^5 to 6.4×10^5 s depending on the family, which falls below one second per problem once that many problems are served, and with 128 problems, which come within 0.04 points of 512 (Table 2), the cost is a quarter of that. Eight attempts per problem instead of 64 give the same schedule in every family at matched wall clock (Table 32) and cut the cost by a further factor of eight, down to 2.36 s per problem over $5{,}000$ served on code.

Table 11: Cost of the calibration at the reference setting: calibration calls and wall clock (s), the number of served problems it is spread over, the cost per served problem (s), the deployment cost (s), and the solve rates on code and math with their mean. The first row estimates (m, n) for each served problem from 128 attempts of its own and is a reference for what per-problem estimation would buy; the other rows use N calibration problems with k attempts of each configuration, and $k = 8$ with $N = 128$ matches the full calibration on code and math.

Configuration	calls	calib. s	spread over	s per problem	deploy s	Code	Math	Avg
Per-problem (m, n), 128 attempts	128	737	1	737.22	23.13	79.45	81.20	80.32
Per family, N=128, k=64	16384	94364	5000	18.87	23.13	75.52	76.69	76.11
Shared by all families, N=512, k=64	65536	377458	20000	18.87	23.13	75.55	76.72	76.13
Per family, N=128, k=32	8192	47182	5000	9.44	23.13	75.55	76.72	76.13
Per family, N=128, k=8	2048	11796	5000	2.36	23.13	75.55	76.72	76.13

Table 12: Cost of the calibration per family, matched wall clock at Qwen2.5-32B: calibration problems, calls (64 attempts per problem and configuration), calibration wall clock (s), the solve rate and deployment cost it yields, and the calibration cost per served problem (s) when spread over 5,000, 20,000 and 100,000 problems.

Family	problems	calls	calib. s	solve	deploy s	5,000	20,000	100,000
Code	64	8,192	47,182	75.55	23.13	9.44	2.36	0.47
	128	16,384	94,364	75.52	23.13	18.87	4.72	0.94
	512	65,536	377,458	75.55	23.13	75.49	18.87	3.77
Math	64	8,192	57,589	76.72	28.22	11.52	2.88	0.58
	128	16,384	115,178	76.69	28.22	23.04	5.76	1.15
	512	65,536	460,710	76.72	28.22	92.14	23.04	4.61
Reasoning	64	8,192	42,580	78.71	28.52	8.52	2.13	0.43
	128	16,384	85,160	81.84	29.14	17.03	4.26	0.85
	512	65,536	340,642	81.87	29.14	68.13	17.03	3.41
Agent	64	8,192	80,007	83.69	53.43	16.00	4.00	0.80
	128	16,384	160,014	83.62	53.43	32.00	8.00	1.60
	512	65,536	640,057	83.69	53.43	128.01	32.00	6.40

E Checks of the Identity, the Estimators and the Pools

E.1 The Identity on Bernoulli Attempts

Propositions 1 and 2 rest on (A1). To check them on sampled attempts, we compared the two-attempt gain that they predict with the gain observed in Bernoulli attempts, over 300 settings (Fig. 4a). The mean difference is -0.0054 and 10 of the 300 settings fall outside the limits of agreement (Bland & Altman, 1986), $+0.120$ and -0.131. When the first-attempt ranking of the two precisions is carried to the fifth attempt, the extrapolated difference falls below the observed one in all 300 settings, with a mean difference of -0.020 (Fig. 4b): the first attempt understates what a switch is worth later on, as the selection effect of Theorem 1 implies. Along the quantiles of the per-problem distribution, the observed and predicted second-attempt solve rates agree on the top quarter of the distribution, where problems are solved most often, and differ below it by at most 0.074, on the math family (Fig. 4c).

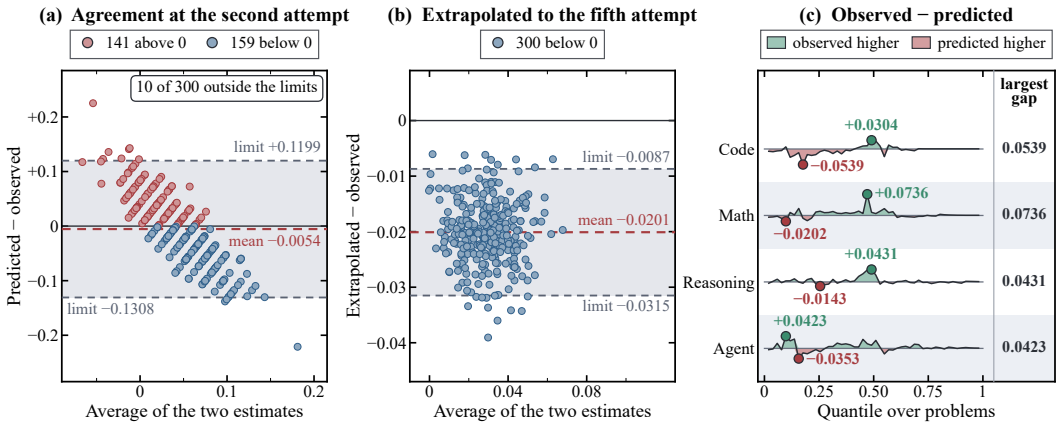


Figure 4: The identity on Bernoulli attempts. (a) Predicted minus observed two-attempt gain against their average for 300 settings (Bland-Altman plot); red points lie above zero and blue points below, the red line is the mean and the band the 95% limits of agreement. (b) The same comparison when the first-attempt ranking is carried to the fifth attempt; all 300 settings lie below zero. (c) Observed minus predicted second-attempt solve rate along the quantiles of the per-problem distribution, one lane per family, filled green where the observed value is larger and red where it is smaller; dots mark the extremes, the right column the largest gap of each lane.

E.2 Simulation Check of the Estimators

Proposition 5 was checked on three constructions whose covariance term is known exactly: independent q_L and q_H, identical $q_H = q_L$, and opposed success probabilities, each with 24 independent replicates of 96 problems and $k \in \{8, 16, 32, 64, 128\}$ attempts per configuration. The plug-in estimate is biased upward in all 15 rows, by 0.020 to 0.027 at $k = 8$ and still by 0.0013 to 0.0015 at $k = 128$, while the split-half bias changes sign across rows and stays within 0.005 of zero (Table 13,

Fig. 5). The split-half estimate also has the lower root mean squared error in every row, by a factor between 1.10 and 3.32, and its sign is never wrong on the constructions where the sign is defined.

Table 13: Simulation check of the estimators of the covariance term $\mathrm{Var}(q_L) - \mathrm{Cov}(q_L, q_H)$ on three constructions with known truth: independent, identical ($q_H = q_L$) and opposed success probabilities, each with 24 replicates of 96 problems for each k. Bias and RMSE of the split-half and plug-in estimators, and the rate of wrong signs of the split-half estimate; the sign is undefined on the identical construction, where the term is zero.

		Bias		RMSE		Sign error
Construction	k	split-half	plug-in	split-half	plug-in	split-half
independent	8	+0.00022	+0.02410	0.01078	0.02559	0
independent	16	−0.00017	+0.01217	0.00562	0.01311	0
independent	32	+0.00072	+0.00655	0.00467	0.00755	0
independent	64	+0.00001	+0.00306	0.00253	0.00389	0
independent	128	−0.00023	+0.00140	0.00164	0.00212	0
identical	8	+0.00059	+0.02682	0.00838	0.02779	–
identical	16	−0.00188	+0.01147	0.00540	0.01221	–
identical	32	+0.00018	+0.00633	0.00228	0.00675	–
identical	64	−0.00026	+0.00290	0.00146	0.00316	–
identical	128	−0.00005	+0.00151	0.00102	0.00180	–
opposed	8	−0.00486	+0.02037	0.01128	0.02201	0
opposed	16	−0.00298	+0.00968	0.00671	0.01145	0
opposed	32	−0.00144	+0.00462	0.00466	0.00612	0
opposed	64	−0.00081	+0.00228	0.00359	0.00394	0
opposed	128	−0.00028	+0.00128	0.00246	0.00283	0

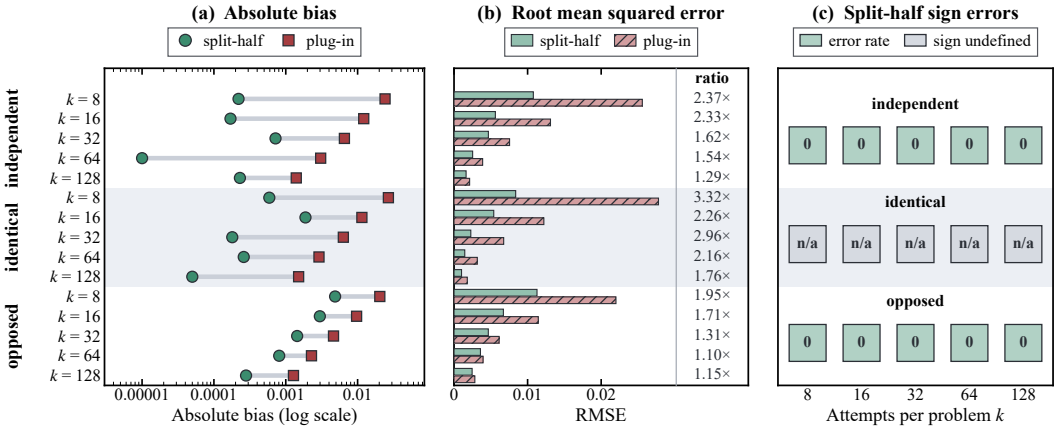


Figure 5: Simulation check of the estimators of Table 13. (a) Absolute bias on a log scale, split-half (circles) and plug-in (squares), for each construction and number of attempts k per configuration. (b) Root mean squared error of both estimators, with the ratio of the plug-in error to the split-half error. (c) Rate of wrong signs of the split-half estimate, over 24 replicates; the sign is undefined on the identical construction, where the term is zero.

E.3 Two-Attempt Diagnostics per Setting

Table 14 reports the terms of Proposition 2 for two attempts on twelve model and recipe settings of the code family. The mean shift $\mu_H - \mu_L$ is 2.3 to 9.4 points, each with $p < 0.05$; $\mathrm{Var}(q_L - q_H)$ is 0.026 to 0.032 in every setting; the two-attempt gain is 2.6 to 7.1 points, with every interval clear of zero; and the covariance term is 1.5 to 2.6 points, between 30% and 60% of the two-attempt gain.

E.4 Joint Law of the Pools

Table 15 gives the joint moments of the Qwen2.5-7B pools, and Tables 16 and 17 and Fig. 6a give the decile tables at Qwen2.5-32B. At the reference setting 144 of the 240 problems lie in the same decile of q_L and q_H, 165 lie within one decile, and 27 of the 100 cells are occupied. The disagreements go both ways: 47 problems have $q_L < 0.2$ and q_H between 0.3 and 0.6, which a switch to H can solve, and 28 have q_L between 0.3 and 0.5 and $q_H < 0.2$, which the low-precision block serves. The four

Table 14: Two-attempt diagnostics for twelve model and recipe settings of the code family (240 problems each). Mean shift $\mu_H - \mu_L$ in points with its z score and sign-permutation p value (399 permutations); $\mathrm{Var}(q_L - q_H)$, and the two-attempt gain $S(1,1) - S(2,0)$ of Proposition 2 in points, with 95% intervals; and the two terms of the gain, the mean term $(1 - \mu_L)(\mu_H - \mu_L)$ and the covariance term $\mathrm{Var}(q_L) - \mathrm{Cov}(q_L, q_H)$, also in points.

		Mean shift			$\mathrm{Var}(q_L - q_H)$			Two-attempt gain			Its two terms	
Model	Recipe	value	z	p	value	low	high	value	low	high	mean	cov.
Qwen2.5-7B	GPTQ-W4	4.00	3.55	0.0025	0.0305	0.0247	0.0363	3.92	2.32	5.59	1.88	2.04
Qwen2.5-7B	AWQ-W4	6.20	5.34	0.0025	0.0323	0.0261	0.0389	5.45	3.86	7.25	3.06	2.39
Qwen2.5-7B	SmoothQuant-W8A8	9.40	8.33	0.0025	0.0306	0.0242	0.0365	7.09	5.31	8.79	4.94	2.15
Qwen2.5-32B	GPTQ-W4	4.05	3.52	0.0025	0.0301	0.0243	0.0368	4.14	2.54	6.09	1.97	2.17
Qwen2.5-32B	AWQ-W4	3.17	2.72	0.0075	0.0301	0.0231	0.0368	3.32	1.67	4.92	1.33	1.99
Qwen2.5-32B	SmoothQuant-W8A8	4.32	3.79	0.0025	0.0295	0.0241	0.0354	4.21	2.46	5.91	1.93	2.28
Qwen2.5-72B	GPTQ-W4	3.80	3.47	0.0050	0.0272	0.0213	0.0329	3.77	2.08	5.35	1.72	2.06
Qwen2.5-72B	AWQ-W4	5.16	4.39	0.0025	0.0294	0.0232	0.0360	4.96	3.50	6.76	2.38	2.58
Qwen2.5-72B	SmoothQuant-W8A8	4.56	3.72	0.0025	0.0323	0.0258	0.0386	4.76	2.93	6.63	2.35	2.40
Mistral-24B	GPTQ-W4	5.05	4.71	0.0025	0.0259	0.0204	0.0317	4.80	3.22	6.47	2.31	2.50
Mistral-24B	AWQ-W4	2.34	2.14	0.0425	0.0294	0.0234	0.0353	2.61	1.02	4.46	1.13	1.48
Mistral-24B	SmoothQuant-W8A8	4.76	4.41	0.0025	0.0292	0.0228	0.0343	4.64	2.86	6.21	2.40	2.24

families at Qwen2.5-32B look alike, with 56% to 60% of the problems in the same decile and 69% to 73% within one, and over all 288 contexts 69.2% of the 28,800 cells of the decile tables are empty.

Table 15: Joint moments of the pool for Qwen2.5-7B: means, variances and covariance of (q_L, q_H) over the 240 evaluation problems of the pool for each family and recipe, and the resulting covariance term $\mathrm{Var}(q_L) - \mathrm{Cov}(q_L, q_H)$. H is the same BF16 model for every recipe, so μ_H and $\mathrm{Var}(q_H)$ do not change within a family.

Family	Recipe	μ_L	μ_H	$\mathrm{Var}(q_L)$	$\mathrm{Var}(q_H)$	$\mathrm{Cov}(q_L, q_H)$	Cov. term
Code	GPTQ-W4	0.529	0.569	0.1470	0.1367	0.1266	0.0204
Code	AWQ-W4	0.507	0.569	0.1522	0.1367	0.1283	0.0239
Code	SmoothQuant-W8A8	0.475	0.569	0.1493	0.1367	0.1277	0.0215
Code	FP8	0.540	0.569	0.1469	0.1367	0.1270	0.0199
Code	QAT-W4	0.512	0.569	0.1461	0.1367	0.1231	0.0230
Code	QuaRot	0.524	0.569	0.1508	0.1367	0.1258	0.0250
Code	SpinQuant	0.522	0.569	0.1406	0.1367	0.1243	0.0163
Code	AQLM / QuIP#	0.495	0.569	0.1408	0.1367	0.1176	0.0232
Math	GPTQ-W4	0.452	0.506	0.1240	0.1153	0.1021	0.0219
Math	AWQ-W4	0.500	0.506	0.1236	0.1153	0.1025	0.0211
Math	SmoothQuant-W8A8	0.465	0.506	0.1304	0.1153	0.1069	0.0235
Math	FP8	0.480	0.506	0.1344	0.1153	0.1182	0.0161
Math	QAT-W4	0.490	0.506	0.1294	0.1153	0.1098	0.0196
Math	QuaRot	0.480	0.506	0.1240	0.1153	0.1066	0.0174
Math	SpinQuant	0.498	0.506	0.1297	0.1153	0.1141	0.0156
Math	AQLM / QuIP#	0.508	0.506	0.1206	0.1153	0.1065	0.0141
Reasoning	GPTQ-W4	0.427	0.500	0.1095	0.0927	0.0833	0.0262
Reasoning	AWQ-W4	0.472	0.500	0.1140	0.0927	0.0952	0.0187
Reasoning	SmoothQuant-W8A8	0.418	0.500	0.1133	0.0927	0.0935	0.0198
Reasoning	FP8	0.468	0.500	0.1153	0.0927	0.0953	0.0200
Reasoning	QAT-W4	0.421	0.500	0.1100	0.0927	0.0893	0.0207
Reasoning	QuaRot	0.479	0.500	0.1010	0.0927	0.0862	0.0148
Reasoning	SpinQuant	0.456	0.500	0.1233	0.0927	0.1003	0.0230
Reasoning	AQLM / QuIP#	0.467	0.500	0.1126	0.0927	0.0922	0.0203
Agent	GPTQ-W4	0.474	0.518	0.1160	0.1085	0.0967	0.0193
Agent	AWQ-W4	0.446	0.518	0.1002	0.1085	0.0791	0.0211
Agent	SmoothQuant-W8A8	0.423	0.518	0.1017	0.1085	0.0858	0.0159
Agent	FP8	0.434	0.518	0.0988	0.1085	0.0843	0.0146
Agent	QAT-W4	0.473	0.518	0.1069	0.1085	0.0839	0.0230
Agent	QuaRot	0.451	0.518	0.1061	0.1085	0.0864	0.0197
Agent	SpinQuant	0.434	0.518	0.0966	0.1085	0.0817	0.0148
Agent	AQLM / QuIP#	0.422	0.518	0.0979	0.1085	0.0823	0.0157

E.5 Survivors After Failures

We follow the problems that stay open as failures of L accumulate (Table 18). After two failures the mean of q_L among them falls from 0.514 to 0.129 and the mean of q_H from 0.554 to 0.237, and the correlation of the two falls from 0.901 to 0.283, so on the survivors the two precisions agree far less than on the full pool. From $r = 4$ on the correlation rises again, to 0.492 at $r = 8$, as the survivors shift toward problems both precisions rarely solve. The right half gives the yields per second behind Fig. 1b and their ratio, which rises at every step and passes one between the first and second failures.

Table 16: Joint decile table of (q_L, q_H) at the reference setting, from the per-problem probabilities of the 240 problems. Rows are bins of width 0.1 of q_L, columns of q_H; each entry counts problems, and the margins give the totals. Of the 240, 144 lie on the diagonal and 165 within one decile of it; 27 of the 100 cells are occupied.

$q_L \setminus q_H$	.0–.1	.1–.2	.2–.3	.3–.4	.4–.5	.5–.6	.6–.7	.7–.8	.8–.9	.9–1	total
.0–.1	32	3	0	9	11	5	0	0	0	0	60
.1–.2	0	8	2	8	11	3	0	0	0	0	32
.2–.3	0	2	4	1	0	0	0	0	0	0	7
.3–.4	9	9	0	0	0	0	0	0	0	0	18
.4–.5	1	9	0	0	0	0	0	0	0	0	10
.5–.6	0	0	0	0	0	1	1	0	0	0	2
.6–.7	0	0	0	0	0	0	3	1	0	0	4
.7–.8	0	0	0	0	0	0	2	8	3	0	13
.8–.9	0	0	0	0	0	0	0	0	3	6	9
.9–1	0	0	0	0	0	0	0	0	0	85	85
total	42	31	6	18	22	9	6	9	6	91	240

Table 17: Decile agreement of the two precisions per family at Qwen2.5-32B with GPTQ-W4: count and share of the 240 problems in the same decile of q_L and q_H and within one decile, and occupied cells of the table.

Family	same decile	share	within one	share	occupied cells
Code	144	60.0%	165	68.8%	27
Math	143	59.6%	173	72.1%	32
Reasoning	141	58.8%	174	72.5%	33
Agent	134	55.8%	170	70.8%	30

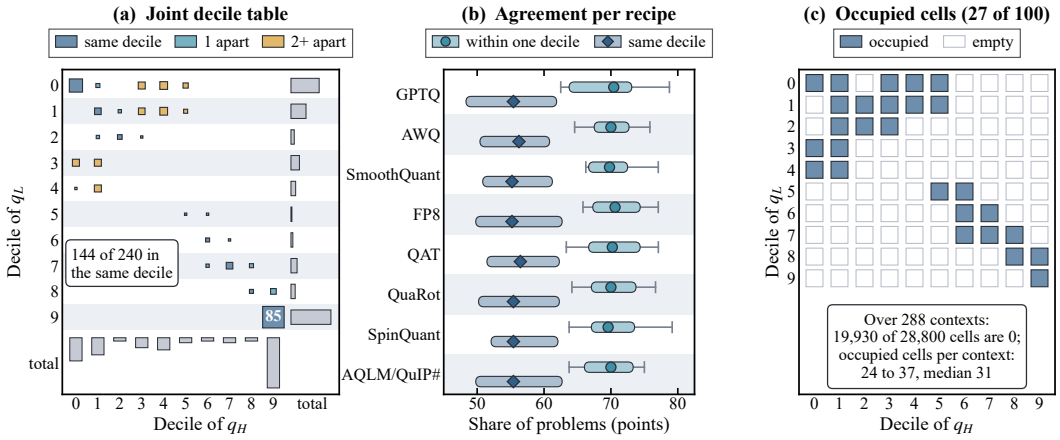


Figure 6: Joint law of (q_L, q_H). (a) Joint decile table at the reference setting; the area of each square is proportional to the number of problems, and its color marks whether q_L and q_H fall in the same decile, one decile apart, or further apart; the bars at the sides give the two marginals. (b) For each recipe over all its contexts, the share of problems within one decile (capsule from the 10th to the 90th percentile, dot at the median, whiskers from the minimum to the maximum) and in the same decile (lower capsule, diamond at the median). (c) Cells that hold at least one problem at the reference setting, and the count of the empty cells over all 288 contexts.

Table 18: Left: survivors at Qwen2.5-32B on code. Means h_r^L and h_r^H of q_L and q_H and their correlation among the problems still open after r failures of L, for GPTQ-W4 and AWQ-W4. **Right: yields per second at the reference setting.** h_r^L/c_L and $h_r^H/(c_H + c_{\mathrm{sw}})$ after r failures of L, their ratio, the expected number of the 240 problems still open, and the expected number of the open problems that one more attempt of L would solve.

Recipe	r	h_r^L	h_r^H	corr.
GPTQ-W4	0	0.514	0.554	0.901
GPTQ-W4	2	0.129	0.237	0.283
GPTQ-W4	4	0.091	0.222	0.269
GPTQ-W4	6	0.073	0.211	0.387
GPTQ-W4	8	0.062	0.198	0.492
AWQ-W4	0	0.512	0.554	0.889
AWQ-W4	2	0.127	0.235	0.291
AWQ-W4	4	0.084	0.218	0.257
AWQ-W4	6	0.065	0.206	0.380
AWQ-W4	8	0.054	0.194	0.489

r	h_r^L/c_L	$h_r^H/(c_H+c_{\mathrm{sw}})$	ratio	open	next L
0	0.1176	0.0765	0.65	240.0	123.32
1	0.0435	0.0380	0.87	116.7	22.20
2	0.0295	0.0328	1.11	94.5	12.17
3	0.0242	0.0315	1.30	82.3	8.69
4	0.0209	0.0307	1.47	73.6	6.72
5	0.0186	0.0303	1.63	66.9	5.42
6	0.0168	0.0291	1.74	61.5	4.50
7	0.0153	0.0286	1.87	57.0	3.81
8	0.0141	0.0273	1.93	53.2	3.28

E.6 Numerical Checks of the Exchange Identity and the Diagnostic Rule

The reweighting sweep of Table 19 holds the means and variances of the code and math pools fixed and moves only $\mathrm{Cov}(q_L, q_H)$. The enumerated two-attempt gain equals $\mathrm{Var} - \mathrm{Cov}$ to within 0.0001 at all eight levels of both pools, as Proposition 2 states when the two means are equal, and the best two-attempt split stays one attempt of each precision. Along a coupling of the calibration pool with all four marginal moments fixed, the covariance term falls linearly from 0.0327 to zero as $\mathrm{Cov} / \mathrm{Var}$ goes from 0 to 1, while the mean term stays zero (Table 20). The diagnostic variant, which chooses between low-precision blocks of 2 and of 6 attempts of L, matches the enumerated optimum in 3 of the 12 code settings, with a mean absolute gap of 1.92 attempts between the two (Fig. 7a, Table 35).

Table 19: Reweighting sweep. Problems of the code and math pools are reweighted so that $\mu_L = \mu_H$ (0.420 and 0.445) and $\mathrm{Var}(q_L) = \mathrm{Var}(q_H)$ (0.0676 and 0.0650), while $\mathrm{Cov}(q_L, q_H)$ moves over eight levels. The two-attempt gain $S(1,1) - S(2,0)$ by exact enumeration on the reweighted pool is compared with $\mathrm{Var} - \mathrm{Cov}$ from Proposition 2; the two agree to 0.0001 at every level, and one attempt of each precision is always optimal.

	Code			Math		
Level	Cov	Var − Cov	gain	Cov	Var − Cov	gain
1	0.0000	0.0676	0.0676	0.0000	0.0650	0.0650
2	0.0093	0.0583	0.0583	0.0089	0.0561	0.0561
3	0.0185	0.0491	0.0491	0.0178	0.0472	0.0472
4	0.0278	0.0398	0.0398	0.0267	0.0383	0.0383
5	0.0371	0.0305	0.0305	0.0357	0.0294	0.0294
6	0.0464	0.0212	0.0213	0.0446	0.0204	0.0204
7	0.0556	0.0120	0.0120	0.0535	0.0115	0.0115
8	0.0649	0.0027	0.0027	0.0624	0.0026	0.0026

Table 20: The covariance term along a fixed-marginal coupling of the calibration pool: twelve couplings with the four marginal moments held fixed and the ratio $\mathrm{Cov} / \mathrm{Var}$ moved from 0 to 1. The covariance term falls linearly to zero, and the mean term is zero in every coupling, since the two means are equal by construction.

Cov / Var	0.000	0.091	0.182	0.273	0.364	0.455	0.545	0.636	0.727	0.818	0.909	1.000
Cov. term	0.0327	0.0297	0.0267	0.0238	0.0208	0.0178	0.0149	0.0119	0.0089	0.0060	0.0030	0.0000
mean term	0	0	0	0	0	0	0	0	0	0	0	0

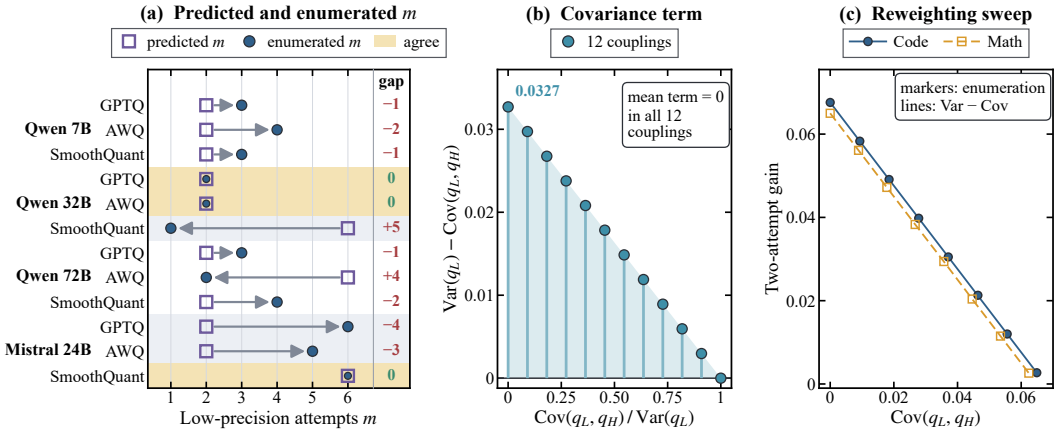


Figure 7: Diagnostic rule and numerical checks of Proposition 2. (a) Low-precision block length m predicted by the eight-attempt diagnostic (open squares) and the enumerated best m (filled circles) in twelve settings of the code family, with the gap. (b) Covariance term $\mathrm{Var}(q_L) - \mathrm{Cov}(q_L, q_H)$ over 12 couplings of the calibration pool with fixed marginal moments, against $\mathrm{Cov} / \mathrm{Var}$. (c) Two-attempt gain by exact enumeration (markers) and $\mathrm{Var} - \mathrm{Cov}$ from Proposition 2 (lines) at the eight covariance levels of the reweighted code and math pools.

F Full Results

F.1 The Reference Setting

Table 21 lists all 21 strategies at the reference setting with bootstrap intervals, paired differences and costs. RetryPrec solves 75.55 points, 3.43 more than fixed L, with a paired interval of $[1.48, 5.62]$, and 4.23 more than fixed H. Four other rules, the fixed and the transferred switch point, the cascade and the diagnostic variant, choose the same pair here and reach the same solve rate; the random pair run interleaved is 0.13 points behind, and temperature and L with a second temperature, at 74.76, are 0.79 points behind with an interval that covers zero. The expected cost of RetryPrec when stopping at the first accepted answer is 11.49 s, against 11.05 s for fixed L and 12.79 s for fixed H at this budget.

Table 21: All strategies at the reference setting (Qwen2.5-32B, GPTQ-W4, code, matched wall clock, $C_{\mathrm{w}} = 24.0$ s). Solve rate in points with a 95% bootstrap interval (2,000 resamples), the paired difference to RetryPrec with its interval, the scheduled cost and the expected cost when stopping at the first accepted answer (in s per problem), the 95th percentile of the cost over problems, and the problems out of 240 it solves in one pass. Grey rows are references. The random pair runs interleaved here; as blocks it equals RetryPrec (Table 1).

	Solve rate			Difference to RetryPrec			Cost (s per problem)			
Strategy	value	low	high	value	low	high	sched.	stop	p95	solved
BF16, one attempt	55.43	50.74	60.25	−20.12	−22.62	−17.78	7.15	7.15	7.15	131
Fixed L	72.12	67.57	76.25	−3.43	−5.62	−1.48	21.85	11.05	21.85	169
Fixed H	71.32	67.01	75.73	−4.23	−5.98	−2.51	21.45	12.79	21.45	174
Oracle, pure	80.05	76.13	83.78	+4.50	+3.62	+5.38	21.73	10.19	21.85	194
Oracle, mixture	80.05	76.14	83.78	+4.50	+3.63	+5.39	21.79	10.22	21.85	193
More samples	72.12	67.57	76.25	−3.43	−5.62	−1.48	21.85	11.05	21.85	169
Temperature T=1.0	74.76	70.50	78.67	−0.79	−2.75	+1.11	22.39	11.01	22.39	178
L+L' (GPTQ, AWQ)	72.53	68.08	76.59	−3.02	−5.23	−0.97	22.39	11.31	22.39	173
L + temperature	74.76	70.50	78.67	−0.79	−2.75	+1.11	22.39	11.01	22.39	178
Random pair, interleaved	75.42	71.50	78.98	−0.13	−1.10	+0.42	23.32	11.49	23.32	180
Fixed m=$\lfloor C/2c_L \rfloor$	75.55	71.86	79.40	0.00	−0.62	+0.58	23.13	11.49	23.13	180
Transferred m	75.55	71.86	79.40	0.00	−0.62	+0.58	23.13	11.49	23.13	180
Cross-config. allocation	72.53	68.08	76.59	−3.02	−5.23	−0.97	22.39	11.31	22.39	173
FrugalGPT cascade	75.55	71.86	79.40	0.00	−0.62	+0.58	23.42	11.78	23.42	180
Hybrid LLM routing	70.77	66.35	75.16	−4.78	−6.78	−2.86	21.87	12.57	22.13	168
RouteLLM	71.06	66.71	75.54	−4.49	−6.58	−2.55	21.91	12.33	22.13	169
Speculative decoding	71.32	67.01	75.73	−4.23	−5.98	−2.51	21.45	12.79	21.45	174
Best-of-n, ORM	72.12	67.57	76.25	−3.43	−5.55	−1.30	22.13	11.34	22.13	169
Best-of-n, PRM	71.32	67.01	75.73	−4.23	−5.98	−2.51	21.73	13.07	21.73	174
RetryPrec	75.55	71.86	79.40	–	–	–	23.13	11.49	23.13	180
RetryPrec, diagnostic	75.55	71.66	79.21	0.00	−0.76	+0.48	23.13	11.49	23.13	180

Under memory-time (Table 22), every schedule that keeps both checkpoints resident is limited to the pair $(2, 1)$ and solves 69.97 points, while the pure low-precision rows afford nineteen attempts of L and solve 86.78 points. The two temperature rows are the highest, at 88.93 and 89.08 points, and lie above the oracle mixture, whose bound covers only schedules over L and H at a single temperature.

F.2 All Strategies at Qwen2.5-32B

Table 23 gives every strategy on the four families of Qwen2.5-32B under both budgets, with the pair each strategy runs at the code setting. At matched wall clock RetryPrec is best or tied for best on code, reasoning and agent; on math, temperature and L with a second temperature are ahead by 1.06 points and the transferred switch point by 0.78. Under memory-time it equals fixed L in every family.

F.3 Equal-Cost Comparison and Budget Sweep

Table 24 lists the thirty settings of Fig. 2, and Fig. 8 shows what the schedule does along the five budgets in all four families. At matched wall clock the low-precision block holds 2 to 6 attempts and the high-precision block 0 to 4; at the smallest budget the enumeration returns the pure low-precision schedule in every family, and it switches at every larger budget. At matched memory-time it returns the pure low-precision schedule in all 20 family and budget settings. The schedule spends 84% to 100% of the allowance, and the diagnostic variant costs a median of 2.7 points at the smallest wall-clock budget and nothing under memory-time, where both variants choose the same schedule.

Table 22: All strategies at the reference setting under the matched memory-time budget (1617.6 GB·s). Grey rows are references. Solve rate in points with its 95% interval, the pair (m, n) (Appendix D.4), the scheduled and the expected cost (s per problem), the peak resident memory (GB) and the memory-time that it uses (GB·s). A strategy that keeps both checkpoints resident holds 86.5 GB during every second of an attempt and affords at most the pair $(2, 1)$; one with L alone holds 19.1 GB and affords up to nineteen attempts of L.

Strategy	value	low	high	(m, n)	sched. s	stop s	peak GB	used GB·s
BF16, one attempt	55.43	50.62	60.24	(0,1)	7.15	7.15	67.4	482
Fixed L	86.78	83.48	90.08	(19,0)	83.02	22.71	19.1	1587
Fixed H	71.32	66.95	75.70	(0,3)	21.45	12.79	67.4	1445
Oracle, pure	87.78	84.56	90.99	(14.35,0.31)	64.95	18.53	67.4	1349
Oracle, mixture	87.78	84.56	90.99	(14.35,0.31)	64.95	18.53	67.4	1349
More samples	86.78	83.48	90.08	(19,0)	83.02	22.71	19.1	1587
Temperature T=1.0	88.93	85.87	91.98	(0,18)	80.61	20.69	19.1	1541
L+L' (GPTQ, AWQ)	87.03	83.74	90.32	(0,18)	80.61	22.29	19.1	1541
L + temperature	89.08	86.14	92.02	(6,13)	84.53	22.03	19.1	1616
Random pair, interleaved	69.97	65.83	74.11	(2,1)	16.08	9.44	86.5	1390
Fixed m=$\lfloor C/2c_L \rfloor$	69.97	65.83	74.11	(2,1)	15.98	9.35	86.5	1382
Transferred m	86.78	83.48	90.08	(19,0)	83.02	22.71	19.1	1587
Cross-config. allocation	87.03	83.74	90.32	(0,18)	80.61	22.29	19.1	1541
FrugalGPT cascade	69.97	65.83	74.11	(2,1)	16.27	9.63	86.5	1407
Hybrid LLM routing	77.80	73.75	81.86	(7.68,1.79)	46.62	16.45	67.4	1516
RouteLLM	78.29	74.25	82.33	(9.90,1.44)	53.80	19.28	67.4	1531
Speculative decoding	71.32	66.95	75.70	(0,3)	21.45	12.79	67.4	1445
Best-of-n, ORM	86.78	83.48	90.08	(19,0)	83.02	22.71	19.1	1587
Best-of-n, PRM	71.32	66.95	75.70	(0,3)	21.73	13.07	67.4	1464
RetryPrec	86.78	83.48	90.08	(19,0)	83.02	22.71	19.1	1587
RetryPrec, diagnostic	86.78	83.48	90.08	(19,0)	83.02	22.71	19.1	1587

Table 23: All strategies at Qwen2.5-32B with GPTQ-W4 under both budgets, in points per family, with the pair (m, n) that each strategy runs at the code setting (Appendix D.4); fractional pairs are means of a per-problem rule, and grey rows are references. Under memory-time no ranked strategy with H resident reaches the pure low-precision rows, and the enumeration returns $n^\star = 0$ in every family, so RetryPrec equals fixed L.

	Matched wall clock				Matched memory-time				(m, n) at code	
Strategy	Code	Math	Reas.	Agent	Code	Math	Reas.	Agent	wall	mem.
BF16, one attempt	55.43	54.60	51.60	54.87	55.43	54.60	51.60	54.87	(0,1)	(0,1)
Fixed L	72.12	76.30	78.71	78.74	86.78	87.96	90.89	91.92	(5,0)	(19,0)
Fixed H	71.32	72.46	75.56	80.41	71.32	72.46	75.56	80.41	(0,3)	(0,3)
Oracle, pure	80.05	81.78	84.73	86.79	87.78	88.57	91.62	93.07	(3.58,0.85)	(14.35,0.31)
Oracle, mixture	80.05	81.78	85.00	86.81	87.78	88.57	91.62	93.07	(3.55,0.88)	(14.35,0.31)
More samples	72.12	76.30	78.71	78.74	86.78	87.96	90.89	91.92	(5,0)	(19,0)
Temperature T=1.0	74.76	77.78	79.41	79.22	88.93	89.73	89.84	90.79	(0,5)	(0,18)
L+L' (GPTQ, AWQ)	72.53	76.36	78.88	78.74	87.03	87.96	91.09	91.92	(0,5)	(0,18)
L + temperature	74.76	77.78	79.44	79.30	89.08	89.85	91.42	92.00	(0,5)	(6,13)
Random pair	75.55	71.56	80.11	79.28	69.97	65.98	78.14	67.35	(2,2)	(2,1)
Fixed m=$\lfloor C/2c_L \rfloor$	75.55	75.06	79.86	81.88	69.97	68.16	75.48	76.77	(2,2)	(2,1)
Transferred m	75.55	77.50	81.87	83.18	86.78	87.96	90.89	89.26	(2,2)	(19,0)
Cross-config. allocation	72.53	76.36	78.88	78.74	87.03	87.96	91.09	91.92	(0,5)	(0,18)
FrugalGPT cascade	75.55	76.72	80.42	83.18	69.97	71.56	77.02	79.43	(2,2)	(2,1)
Hybrid LLM routing	70.77	72.69	76.16	79.31	77.80	77.76	81.50	84.84	(1.71,1.98)	(7.68,1.79)
RouteLLM	71.06	73.49	77.12	79.42	78.29	80.11	81.34	87.01	(2.27,1.64)	(9.90,1.44)
Speculative decoding	71.32	72.46	75.56	80.41	71.32	72.46	75.56	80.41	(0,3)	(0,3)
Best-of-n, ORM	72.12	76.30	78.71	78.74	86.78	87.96	90.89	91.92	(5,0)	(19,0)
Best-of-n, PRM	71.32	72.46	75.56	80.41	71.32	72.46	75.56	80.41	(0,3)	(0,3)
RetryPrec	75.55	76.72	81.87	83.69	86.78	87.96	90.89	91.92	(2,2)	(19,0)
RetryPrec, diagnostic	75.55	76.72	80.42	83.18	86.78	87.96	90.89	91.92	(2,2)	(19,0)

Table 24: Equal-cost comparison of Fig. 2, all 30 settings: solve rate in points of nine strategies at Qwen2.5-32B with GPTQ-W4, for code, math and reasoning at five budgets under each budget basis (0.55, 0.78, 1, 1.35 and 1.8 times the nominal budget). Budgets are in s for wall clock and in GB·s for memory-time. Bold marks the best and underline the second best value of each row among the nine strategies; the oracle mixture is a reference.

Family	Basis	Budget	Fixed L	Fixed H	More samp.	Temp.	$L{+}L'$	Rand. pair	Cascade	Specul.	RetryPrec	Oracle mix.
Code	wall	13.2	**65.70**	55.43	**65.70**	62.31	**65.70**	64.77	64.77	55.43	**65.70**	69.92
		18.7	69.32	65.73	69.32	**71.75**	69.66	64.77	64.77	65.73	69.97	75.72
		24.0	72.12	71.32	72.12	74.76	72.53	**75.55**	**75.55**	71.32	**75.55**	80.05
		32.4	76.26	75.06	76.26	79.14	76.77	75.55	79.15	75.06	**80.31**	83.68
		43.2	79.21	79.88	79.21	82.17	79.81	81.37	**83.48**	79.88	**83.48**	86.70
Code	mem	889.7	80.40	55.43	80.40	**83.36**	81.03	76.26	51.38	55.43	80.40	80.60
		1261.7	84.63	65.73	84.63	**86.79**	84.66	64.77	64.77	65.73	84.63	85.27
		1617.6	86.78	71.32	86.78	**88.93**	87.03	69.97	69.97	71.32	86.78	87.78
		2183.8	89.21	75.06	89.21	**91.16**	89.56	69.97	75.55	75.06	89.21	90.00
		2911.7	90.89	79.88	90.89	**92.59**	91.18	64.77	78.29	79.88	90.89	91.69
Math	wall	15.8	68.16	54.60	68.16	**69.18**	68.28	65.98	65.98	54.60	68.16	71.49
		22.5	71.70	66.62	71.70	**72.90**	71.79	65.98	65.98	66.62	71.56	76.83
		28.8	76.30	72.46	76.30	**77.78**	76.36	71.56	76.72	72.46	76.72	81.78
		38.9	79.33	76.14	79.33	80.97	79.37	71.56	79.29	76.14	**81.14**	84.64
		51.8	82.46	78.75	82.46	83.28	82.49	83.53	82.06	78.75	**84.64**	86.99
Math	mem	1067.6	82.46	54.60	82.46	**84.22**	82.46	62.70	51.37	54.60	82.46	82.55
		1514.1	85.80	66.62	85.80	**87.60**	85.80	65.98	65.98	66.62	85.80	86.23
		1941.1	87.96	72.46	87.96	**89.73**	87.96	65.98	71.56	72.46	87.96	88.57
		2620.5	90.19	76.14	90.19	**91.64**	90.19	72.88	75.06	76.14	90.19	90.70
		3494.0	91.91	78.75	91.91	**93.31**	91.91	75.06	79.29	78.75	91.91	92.28
Reasoning	wall	16.5	**72.01**	65.28	**72.01**	69.11	**72.01**	71.60	65.16	65.28	**72.01**	77.30
		23.4	74.78	71.67	74.78	75.66	74.89	65.16	77.02	71.67	**78.14**	81.86
		30.0	78.71	75.56	78.71	79.41	78.88	80.11	80.42	75.56	**81.87**	85.00
		40.4	81.47	80.34	81.47	81.97	81.71	65.16	84.64	80.34	**85.51**	87.58
		53.9	85.26	83.41	85.26	84.61	85.62	82.03	87.25	83.41	**88.12**	89.77
Reasoning	mem	1112.1	**85.98**	65.28	**85.98**	85.30	**85.98**	65.16	65.16	65.28	**85.98**	86.64
		1577.2	**89.17**	71.67	**89.17**	88.15	**89.17**	71.60	71.60	71.67	**89.17**	89.88
		2022.0	90.89	75.56	90.89	89.84	**91.09**	78.14	77.02	75.56	90.89	91.62
		2729.7	92.88	80.34	92.88	91.37	**93.09**	81.65	79.86	80.34	92.88	93.52
		3639.6	94.41	83.41	94.41	92.49	**94.50**	77.04	84.30	83.41	94.41	94.84

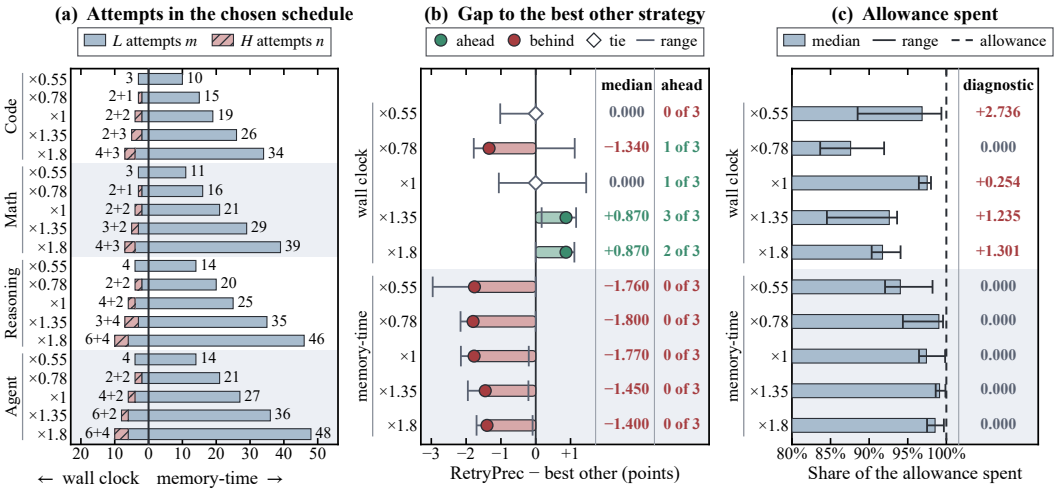


Figure 8: Budget sweep at Qwen2.5-32B, five multiples of the nominal budget. (a) Numbers m and n of low- and high-precision attempts chosen per family at matched wall clock (left) and matched memory-time (right). (b) RetryPrec minus the best other strategy of Table 24: median over three families (dot and bar) with their range (whisker), and the number of families in which RetryPrec is ahead. (c) Share of the allowance that the chosen schedule spends (median as bar, range as whisker), and the median solve rate that the diagnostic variant gives up.

F.4 Models, Recipes, Verifiers and Retry Mechanisms

We run RetryPrec along every axis of the setup at matched wall clock (Table 25). Across nine models the four-family average lies between 73.33 and 79.46 points and across eight recipes between 76.27 and 79.46. The visible-test verifier lowers the average by 0.47 points. Retries with feedback reach 82.59, but they pass information from the failed attempt to the next and leave the setting of (A1); they serve as a control. Table 26 compares RetryPrec with the best other strategy of each of the 36 model and family settings: 33 of the 36 intervals cover zero, one favors RetryPrec and two the other strategy.

Table 25: RetryPrec across models, recipes, verifiers and retry mechanisms, matched wall clock: solve rate in points per family, the four-family average, and the switch charge c_{sw} in ms. **Left:** nine models with GPTQ-W4 as their L. **Right:** eight recipes on Qwen2.5-32B, the two verifiers, and the retry mechanism; the grey row runs Self-Refine and Reflexion, which pass feedback between attempts and so break (A1), as a control.

Model	Code	Math	Reas.	Agent	Avg	ms
Qwen2.5-7B	75.86	78.55	81.05	81.18	79.16	176
Qwen2.5-14B	82.26	78.06	78.11	75.73	78.54	125
Qwen2.5-32B	75.55	76.72	81.87	83.69	79.46	94
Qwen2.5-72B	80.34	75.27	80.25	75.07	77.73	123
Llama-3.1-8B	73.40	76.79	77.77	77.91	76.47	206
Llama-3.1-70B	74.40	73.90	74.80	70.23	73.33	56
R1-Distill-Qwen-7B	76.68	71.24	82.15	72.00	75.52	114
R1-Distill-Qwen-32B	76.14	76.62	71.65	73.73	74.53	63
Mistral-Small-24B	74.96	77.62	74.31	78.31	76.30	85

Setting	Code	Math	Reas.	Agent	Avg	ms
GPTQ-W4	75.55	76.72	81.87	83.69	79.46	94
AWQ-W4	81.95	79.66	77.02	75.25	78.47	94
SmoothQuant-W8A8	79.46	74.85	76.02	78.79	77.28	94
FP8	76.90	78.30	74.09	82.56	77.96	94
QAT-W4	79.14	74.54	78.82	72.57	76.27	94
QuaRot	77.99	77.68	80.00	76.45	78.03	94
SpinQuant	80.28	72.43	79.76	78.44	77.73	94
AQLM / QuIP#	77.58	74.47	82.52	82.08	79.16	94
Perfect verifier	75.55	76.72	81.87	83.69	79.46	94
Visible-test verifier	75.23	76.39	81.29	83.04	78.99	94
Independent restarts	75.55	76.72	81.87	83.69	79.46	94
Retries with feedback	77.97	79.01	85.92	87.44	82.59	94

Table 26: Paired differences per model and family, matched wall clock: RetryPrec minus the best other strategy of each of the 36 settings, named in column three, in points with the 95% interval. 33 of the 36 intervals cover zero, one lies above it and two below. R1-Qwen stands for DeepSeek-R1-Distill-Qwen, Mistral-24B for Mistral-Small-24B, Rand. int. for the random pair run interleaved, and Transf. m for the transferred switch point.

Model	Family	Best other	diff.	low	high
Qwen2.5-7B	Code	$L{+}L'$	−0.06	−0.61	+0.87
Qwen2.5-7B	Math	Fixed m	+0.00	−1.11	+1.08
Qwen2.5-7B	Reas.	Cascade	+0.00	−1.39	+0.85
Qwen2.5-7B	Agent	Cascade	+1.05	−0.12	+2.11
Qwen2.5-14B	Code	Fixed m	+0.04	−1.21	+1.11
Qwen2.5-14B	Math	Rand. int.	−0.01	−1.13	+0.58
Qwen2.5-14B	Reas.	Temp.	−0.18	−1.23	+1.06
Qwen2.5-14B	Agent	Cascade	+0.07	−1.36	+0.79
Qwen2.5-32B	Code	Fixed m	+0.00	−0.58	+0.62
Qwen2.5-32B	Math	Temp.	−1.06	−1.77	+0.41
Qwen2.5-32B	Reas.	Transf. m	+0.00	−0.56	+0.83
Qwen2.5-32B	Agent	Transf. m	+0.51	−0.56	+1.64
Qwen2.5-72B	Code	Fixed m	+0.00	−0.88	+1.30
Qwen2.5-72B	Math	Diagnostic	+0.00	−0.81	+0.61
Qwen2.5-72B	Reas.	Fixed m	+0.00	−1.14	+1.24
Qwen2.5-72B	Agent	Temp.	−0.20	−1.60	+1.24
Llama-3.1-8B	Code	Cascade	+0.27	−0.47	+1.75
Llama-3.1-8B	Math	$L{+}L'$	−0.07	−1.51	+1.20
Llama-3.1-8B	Reas.	Transf. m	−0.01	−1.06	+0.74
Llama-3.1-8B	Agent	Temp.	+0.90	+0.23	+1.98
Llama-3.1-70B	Code	L+temp.	−0.44	−1.65	+0.50
Llama-3.1-70B	Math	Temp.	−0.12	−1.12	+0.64
Llama-3.1-70B	Reas.	L+temp.	−0.69	−1.80	+0.30
Llama-3.1-70B	Agent	Temp.	−1.33	−2.16	−0.39
R1-Qwen-7B	Code	Cascade	+0.14	−0.80	+1.52
R1-Qwen-7B	Math	Rand. int.	+0.00	−0.62	+0.95
R1-Qwen-7B	Reas.	Fixed m	−0.01	−0.57	+1.15
R1-Qwen-7B	Agent	L+temp.	−1.13	−2.61	−0.29
R1-Qwen-32B	Code	Temp.	−0.37	−1.80	+0.88
R1-Qwen-32B	Math	Temp.	−0.40	−1.80	+0.46
R1-Qwen-32B	Reas.	Temp.	−0.62	−1.97	+0.28
R1-Qwen-32B	Agent	L+temp.	−0.48	−1.58	+0.37
Mistral-24B	Code	$L{+}L'$	−0.55	−1.71	+0.11
Mistral-24B	Math	Temp.	−1.07	−1.98	+0.10
Mistral-24B	Reas.	Transf. m	+0.04	−0.85	+1.51
Mistral-24B	Agent	Fixed m	−0.04	−1.06	+1.24

F.5 Crossing Point Across Contexts

Fig. 9 extends Fig. 1b to all 288 contexts. In the median context the switch overtakes one more attempt of L after three failures, with 80.3 of the 240 problems still open. Over the contexts the first crossing comes after two failures in 20.5% of them, after three in 37.8%, after four in 21.2%, and after at most eleven in all of them. The share of contexts in which the switch wins never falls with more failures, and the switch charge lowers the yield of the switch by at most 0.00051 per second.

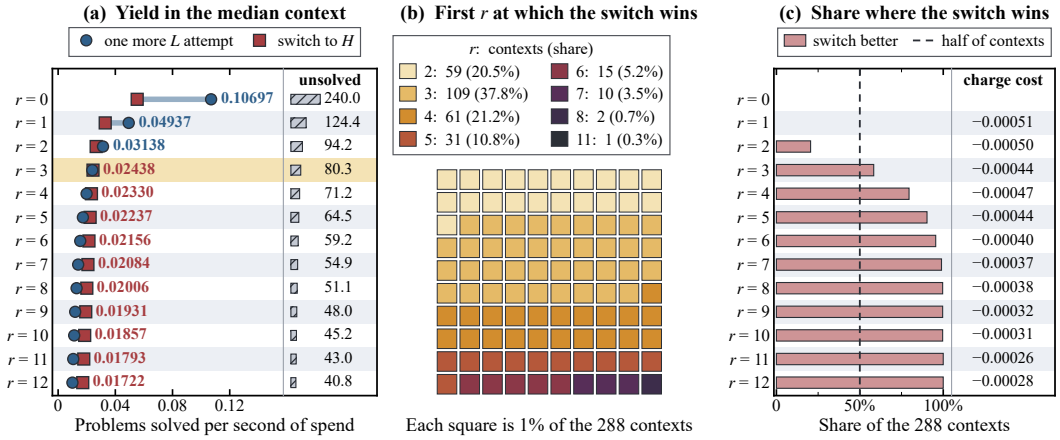


Figure 9: When the switch overtakes one more low-precision attempt, over 288 contexts. (a) In the median context, the yield per second of one more attempt of L (circles) and of a switch to H with its charge (squares) after r failures of L; the number printed is the larger of the two, the right column counts the problems that are still open, and the yellow row marks the first r at which the switch wins. (b) First number of failures at which the switch wins; each square is 1% of the contexts. (c) Share of contexts in which the switch has the higher yield after r failures, with the change in the yield of the switch that its charge causes from $r = 1$ on (right column).

G Complete Ablation Studies

This appendix gives the ablations of Table 2 per family, under both budgets where both apply, with their costs and pairs (m, n). Unless noted, rows use Qwen2.5-32B with GPTQ-W4 as L, and the switch charge of 0.094 s per executed change of precision is included in every cost that we report.

G.1 Switch Point

In expectation over the draw, a pair drawn uniformly from the feasible ones and run interleaved loses 7.5 to 9.9 points against the calibrated pair at matched wall clock and 11.2 to 12.4 points at matched memory-time, and it spends less of the budget, because the draw often returns a pair with few attempts (Table 27). We trace the single draw of Table 2, run interleaved, from fixed L to RetryPrec in Table 28. The drawn pair helps when it has two high-precision attempts, as on code, and helps less or even hurts when it has one; replacing it by the calibrated pair adds 0.13 to 5.04 points per family.

Table 27: Calibrated against random switch points at Qwen2.5-32B, both budgets. The random rule draws one feasible pair uniformly and runs it interleaved; its costs and (m, n) are means over the random draw, and the last column counts the feasible pairs of a setting, from 13 on code at wall clock to 37 on agent at memory-time.

Family	Budget	Rule	solve	sched. s	stop s	m	n	feasible
Code	wall	random	66.70	15.25	9.23	1.85	1.00	13
Code	wall	calibrated	75.55	23.13	11.49	2	2	13
Code	mem	random	74.76	37.53	14.16	8.04	0.33	24
Code	mem	calibrated	86.78	83.02	22.71	19	0	24
Math	wall	random	69.19	19.39	10.96	2.27	0.93	15
Math	wall	calibrated	76.72	28.22	13.15	2	2	15
Math	mem	random	76.78	44.94	15.71	9.00	0.31	26
Math	mem	calibrated	87.96	98.11	24.43	21	0	26
Reasoning	wall	random	72.37	19.27	9.97	2.48	1.43	23
Reasoning	wall	calibrated	81.87	29.14	12.22	4	2	23
Reasoning	mem	random	78.47	42.10	13.81	9.42	0.58	36
Reasoning	mem	calibrated	90.89	101.87	22.29	25	0	36
Agent	wall	random	73.82	34.55	17.76	2.55	1.32	22
Agent	wall	calibrated	83.69	53.43	21.91	4	2	22
Agent	mem	random	80.20	81.32	25.26	10.57	0.49	37
Agent	mem	calibrated	91.92	192.30	39.73	27	0	37

Table 28: From fixed L to RetryPrec, Qwen2.5-32B, matched wall clock. Solve rate of fixed L; the pair drawn for the family and its change when run interleaved; the further change when it is replaced by the calibrated pair; and the result with that pair. On code the draw returns the calibrated pair, and the second step runs it as blocks.

Family	Fixed L	drawn (m, n)	random pair	calibrated pair	RetryPrec	calibrated (m, n)
Code	72.12	(2,2)	+3.30	+0.13	75.55	(2,2)
Math	76.30	(2,1)	−4.62	+5.04	76.72	(2,2)
Reasoning	78.71	(5,1)	+1.53	+1.63	81.87	(4,2)
Agent	78.74	(4,1)	+0.61	+4.34	83.69	(4,2)

G.2 Partner of the Low-Precision Copy

We change the coupling of the pool and recalibrate the schedule each time (Table 29, Fig. 10). With q_H replaced by q_L the enumeration returns the pure low-precision schedule in every family; with the means aligned it gains 0 to 1.9 points over it; as measured it gains 0.42 to 4.95; and with anti-correlated failures it reaches 98.8 to 99.3. Under memory-time the measured pool never pays for the second checkpoint, but the anti-correlated pool does: the enumeration returns $(2, 1)$ or $(2, 2)$, solves 95.8 to 98.1 points, and cuts the scheduled cost to about a fifth of that of pure low precision.

Table 29: Four couplings of the pool, Qwen2.5-32B, with the solve rate, the scheduled and expected cost (s) and the pair (m, n) of the recalibrated schedule: the pool as it was measured; reweighted so that the two means coincide; with anti-correlated failures; and with $q_H = q_L$ on every problem. Under memory-time the couplings with aligned means and with $q_H = q_L$ give the pure low-precision schedule of the measured pool and are omitted. The last column gives the covariance term $\mathrm{Var}(q_L) - \mathrm{Cov}(q_L, q_H)$ of each of the four couplings.

Family	Budget	Coupling	solve	sched. s	stop s	m	n	Cov. term
Code	wall	as measured	75.55	23.13	11.49	2	2	0.0217
Code	wall	means aligned	72.56	23.13	11.64	2	2	0.0078
Code	wall	anti-correlated	98.94	23.13	9.59	2	2	0.3051
Code	wall	identical q	72.12	21.85	11.05	5	0	0
Code	mem	as measured	86.78	83.02	22.71	19	0	0.0217
Code	mem	anti-correlated	96.63	15.98	9.35	2	1	0.3051
Math	wall	as measured	76.72	28.22	13.15	2	2	0.0171
Math	wall	means aligned	76.30	28.03	12.70	6	0	0.0051
Math	wall	anti-correlated	98.82	28.22	10.88	2	2	0.2669
Math	wall	identical q	76.30	28.03	12.70	6	0	0
Math	mem	as measured	87.96	98.11	24.43	21	0	0.0171
Math	mem	anti-correlated	95.80	18.83	10.48	2	1	0.2669
Reasoning	wall	as measured	81.87	29.14	12.22	4	2	0.0173
Reasoning	wall	means aligned	78.71	28.52	12.13	7	0	0.0027
Reasoning	wall	anti-correlated	99.29	29.14	11.06	4	2	0.2306
Reasoning	wall	identical q	78.71	28.52	12.13	7	0	0
Reasoning	mem	as measured	90.89	101.87	22.29	25	0	0.0173
Reasoning	mem	anti-correlated	97.78	20.99	9.08	2	2	0.2306
Agent	wall	as measured	83.69	53.43	21.91	4	2	0.0238
Agent	wall	means aligned	80.65	53.43	22.18	4	2	0.0123
Agent	wall	anti-correlated	99.32	53.43	19.77	4	2	0.2237
Agent	wall	identical q	78.74	49.86	21.26	7	0	0
Agent	mem	as measured	91.92	192.30	39.73	27	0	0.0238
Agent	mem	anti-correlated	98.08	39.18	16.39	2	2	0.2237

Temperature is a separate lever, because it changes the success probabilities themselves. In Table 31 the whole budget goes to one configuration at $T = 0.7$ or at $T = 1.3$: at $T = 1.3$ the solve rate is higher than at $T = 0.7$ in every family and budget. RetryPrec takes the two configurations as given: it runs L and H at $T = 0.7$, as do all other strategies except the temperature comparators at $T = 1.0$.

G.3 Order and Number of Switches

With the same pair, running the high-precision block first raises the expected cost by 1.5 to 3.4 s (Table 30), as Proposition 6 predicts from the yields per second. Interleaving the same attempts adds two or three switch charges, 0.19 to 0.28 s of scheduled cost and up to 2.7 s of expected cost (Table 31). The solve rate, which Proposition 1 makes independent of the order, moves by at most 0.16 points on the sampled attempts, and a second switch moves it by 0.06 to 0.08 points (Table 32).

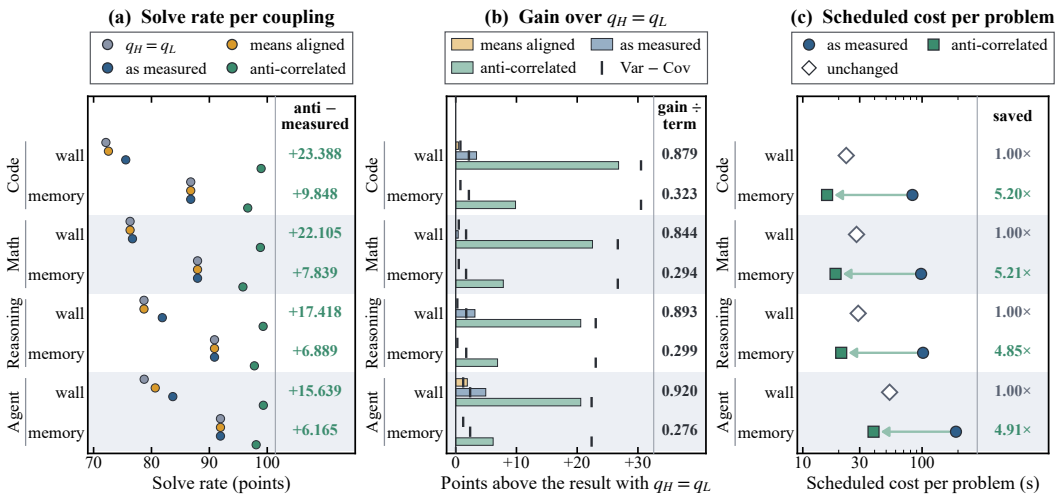


Figure 10: Four couplings of the pool (Table 29). (a) Solve rate of the recalibrated schedule per family and budget with q_H replaced by q_L ($q_H = q_L$), means aligned, as measured, and anti-correlated failures, with the difference of the last two. (b) Points above the result with $q_H = q_L$ for each coupling (bars), with the covariance term $\times 100$ on the same scale (ticks) and their ratio for the anti-correlated pool. (c) Scheduled cost per problem as measured (circles) and anti-correlated (squares), with the factor saved; diamonds mark an unchanged cost.

Table 30: Order of the two blocks at the same pair (m, n), Qwen2.5-32B. With H first the expected stop cost is 1.5 to 3.4 s higher, and the solve rate stays within 0.16 points. Under memory-time the schedule has one block, so the question of order does not arise. The last column is the covariance term of the pool of the family.

Family	Budget	Order	solve	sched. s	stop s	m	n	Cov. term
Code	wall	L then H	75.55	23.13	11.49	2	2	0.0217
Code	wall	H then L	75.47	23.13	13.10	2	2	0.0217
Math	wall	L then H	76.72	28.22	13.15	2	2	0.0171
Math	wall	H then L	76.58	28.22	16.51	2	2	0.0171
Reasoning	wall	L then H	81.87	29.14	12.22	4	2	0.0173
Reasoning	wall	H then L	81.71	29.14	13.73	4	2	0.0173
Agent	wall	L then H	83.69	53.43	21.91	4	2	0.0238
Agent	wall	H then L	83.55	53.43	24.68	4	2	0.0238
Code	mem	one block	86.78	83.02	22.71	19	0	0.0217
Math	mem	one block	87.96	98.11	24.43	21	0	0.0171
Reasoning	mem	one block	90.89	101.87	22.29	25	0	0.0173
Agent	mem	one block	91.92	192.30	39.73	27	0	0.0238

Table 31: Blocked against interleaved execution, and temperature as the only resource, Qwen2.5-32B. Interleaved rows alternate the two precisions with the same (m, n), L first, and pay c_{sw} at every change, so their scheduled cost is 0.19 to 0.28 s higher at wall clock. Temperature rows spend the whole budget on one configuration, at $T = 0.7$ or at $T = 1.3$, and there n counts the attempts at $T = 1.3$. Costs are in s per problem.

Family	Budget	Variant	solve	sched.	stop	m	n
Code	wall	blocked	75.55	23.13	11.49	2	2
Code	wall	interleaved	75.42	23.32	11.49	2	2
Code	wall	T=0.7	72.12	21.85	11.05	5	0
Code	wall	T=1.3	77.51	22.39	10.77	0	5
Code	mem	blocked	86.78	83.02	22.71	19	0
Code	mem	interleaved	86.78	83.02	22.71	19	0
Code	mem	T=0.7	86.78	83.02	22.71	19	0
Code	mem	T=1.3	91.46	80.61	18.82	0	18
Math	wall	blocked	76.72	28.22	13.15	2	2
Math	wall	interleaved	76.72	28.41	15.84	2	2
Math	wall	T=0.7	76.30	28.03	12.70	6	0
Math	wall	T=1.3	81.36	28.73	12.31	0	6
Math	mem	blocked	87.96	98.11	24.43	21	0
Math	mem	interleaved	87.96	98.11	24.43	21	0
Math	mem	T=0.7	87.96	98.11	24.43	21	0
Math	mem	T=1.3	92.86	100.56	20.56	0	21

Family	Budget	Variant	solve	sched.	stop	m	n
Reasoning	wall	blocked	81.87	29.14	12.22	4	2
Reasoning	wall	interleaved	81.87	29.42	13.73	4	2
Reasoning	wall	T=0.7	78.71	28.52	12.13	7	0
Reasoning	wall	T=1.3	84.24	29.24	11.56	0	7
Reasoning	mem	blocked	90.89	101.87	22.29	25	0
Reasoning	mem	interleaved	90.89	101.87	22.29	25	0
Reasoning	mem	T=0.7	90.89	101.87	22.29	25	0
Reasoning	mem	T=1.3	95.21	104.42	18.09	0	25
Agent	wall	blocked	83.69	53.43	21.91	4	2
Agent	wall	interleaved	83.69	53.71	24.16	4	2
Agent	wall	T=0.7	78.74	49.86	21.26	7	0
Agent	wall	T=1.3	84.45	51.10	20.21	0	7
Agent	mem	blocked	91.92	192.30	39.73	27	0
Agent	mem	interleaved	91.92	192.30	39.73	27	0
Agent	mem	T=0.7	91.92	192.30	39.73	27	0
Agent	mem	T=1.3	95.75	189.81	31.45	0	26

Table 32: Left: attempts per calibration problem k. The chosen schedule and its solve rate for k from 8 to 128 (32 to 128 under memory-time, where m reaches 27); only the math family at wall clock moves, between $(2, 2)$ and $(4, 1)$. **Right: a second switch, and residency of H.** Allowing a second switch leaves the pair unchanged and moves the solve rate by 0.06 to 0.08 points; reloading H from host memory at 24 GB/s instead of keeping it resident shortens the high-precision block and thereby lowers the solve rate by 1.66 to 2.49 points.

Family	Budget	k	solve	m	n
Code	wall	8–128	75.55	2	2
Code	mem	32–128	86.78	19	0
Math	wall	8, 16, 64	76.72	2	2
Math	wall	32, 128	77.50	4	1
Math	mem	32–128	87.96	21	0
Reasoning	wall	8–128	81.87	4	2
Reasoning	mem	32–128	90.89	25	0
Agent	wall	8–128	83.69	4	2
Agent	mem	32–128	91.92	27	0

Family	Budget	Variant	solve	sched. s	stop s	m	n
Code	wall	two switches	75.61	23.13	11.49	2	2
Math	wall	two switches	76.80	28.22	13.15	2	2
Reasoning	wall	two switches	81.93	29.14	12.22	4	2
Agent	wall	two switches	83.76	53.43	21.91	4	2
Code	wall	resident	75.55	23.13	11.49	2	2
Code	wall	reload	73.52	23.16	11.66	3	1
Math	wall	resident	76.72	28.22	13.15	2	2
Math	wall	reload	75.06	26.31	12.60	3	1
Reasoning	wall	resident	81.87	29.14	12.22	4	2
Reasoning	wall	reload	80.11	29.70	12.51	5	1
Agent	wall	resident	83.69	53.43	21.91	4	2
Agent	wall	reload	81.20	50.95	21.70	5	1

G.4 Residency and Serving Methods

Reloading H from host memory at 24 GB/s instead of keeping it resident makes the enumeration shorten the high-precision block to a single attempt in every family, and the solve rate falls by 1.66 to 2.49 points (Table 32). Speculative decoding, KV-cache quantization and prefix caching lower the unit costs, and the enumeration may move the chosen pair with them (Table 33). With speculative decoding it takes one or two more high-precision attempts and gains 0.33 to 1.35 points at matched wall clock. With prefix caching it gains up to 0.82 points, except on the agent family, where it moves to a longer low-precision block and loses 0.96 points. Under memory-time, KV-cache quantization and prefix caching let one to three more attempts of L fit into the budget and add 0.23 to 0.83 points.

Table 33: Left: accelerations added to the schedule at Qwen2.5-32B. Speculative decoding, KV-cache quantization (Liu et al., 2024b) and prefix caching change the unit costs of the attempts, and the enumeration may then choose a different pair (m, n). **Right: the perfect and the visible-test verifier.** The visible-test verifier accepts about 0.73% of wrong answers; the rate per model is in Table 10. All costs are in s per problem.

Family	Budget	Component	solve	sched.	stop	m	n
Code	wall	none	75.55	23.13	11.49	2	2
Code	wall	speculative	76.00	22.36	11.11	1	3
Code	wall	KV quant.	75.55	23.13	11.49	2	2
Code	wall	prefix cache	76.14	22.91	11.02	4	1
Code	mem	none	86.78	83.02	22.71	19	0
Code	mem	KV quant.	87.21	87.39	23.29	20	0
Code	mem	prefix cache	87.61	84.14	21.87	21	0
Math	wall	none	76.72	28.22	13.15	2	2
Math	wall	speculative	77.05	28.07	13.25	1	3
Math	wall	prefix cache	76.72	27.29	12.65	2	2
Math	mem	KV quant.	88.31	102.78	24.99	22	0
Math	mem	prefix cache	88.31	97.66	23.74	22	0
Reasoning	wall	none	81.87	29.14	12.22	4	2
Reasoning	wall	speculative	82.75	29.46	12.00	2	4
Reasoning	wall	prefix cache	82.69	29.15	11.63	3	3
Reasoning	mem	KV quant.	91.17	105.94	22.66	26	0
Reasoning	mem	prefix cache	91.68	103.67	21.24	28	0
Agent	wall	none	83.69	53.43	21.91	4	2
Agent	wall	speculative	85.04	52.29	21.09	3	3
Agent	wall	prefix cache	82.73	52.57	21.37	6	1
Agent	mem	KV quant.	92.15	199.42	40.30	28	0
Agent	mem	prefix cache	92.15	188.66	38.13	28	0

Family	Budget	Verifier	solve	stop	m	n
Code	wall	perfect	75.55	11.49	2	2
Code	wall	visible	75.23	11.39	2	2
Code	mem	perfect	86.78	22.71	19	0
Code	mem	visible	85.91	22.71	19	0
Math	wall	perfect	76.72	13.15	2	2
Math	wall	visible	76.39	13.03	2	2
Math	mem	perfect	87.96	24.43	21	0
Math	mem	visible	87.06	24.43	21	0
Reasoning	wall	perfect	81.87	12.22	4	2
Reasoning	wall	visible	81.29	12.05	4	2
Reasoning	mem	perfect	90.89	22.29	25	0
Reasoning	mem	visible	89.64	22.29	25	0
Agent	wall	perfect	83.69	21.91	4	2
Agent	wall	visible	83.04	21.59	4	2
Agent	mem	perfect	91.92	39.73	27	0
Agent	mem	visible	90.18	39.73	27	0

The visible-test verifier accepts about 0.73% of wrong answers. It costs RetryPrec 0.31 to 0.65 points at matched wall clock and 0.87 to 1.74 points at matched memory-time over three models and four families, more under memory-time because the schedule runs more attempts there (Fig. 11). Every strategy that retries loses under this verifier; a single BF16 attempt never retries and loses nothing.

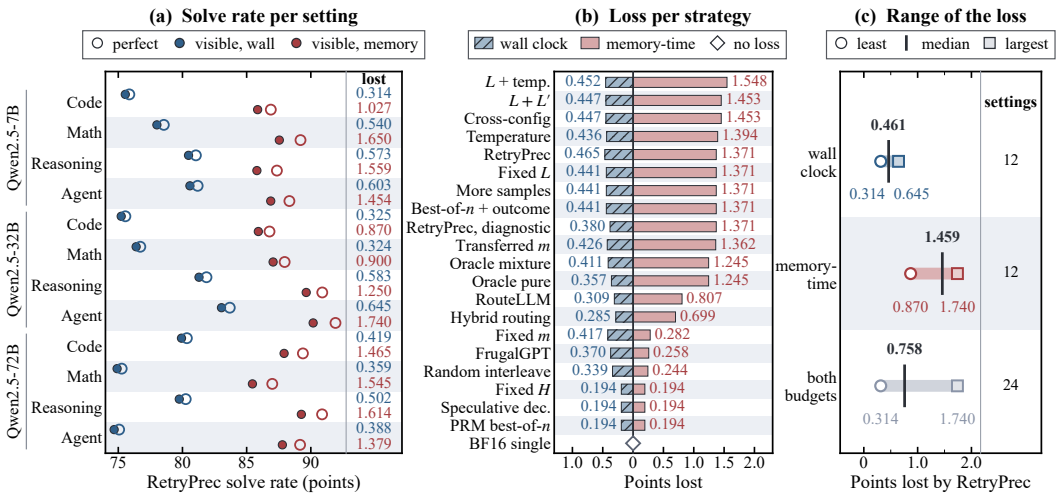


Figure 11: The visible-test verifier. (a) RetryPrec with the perfect verifier (open circles) and with visible tests (filled; blue for wall clock, red for memory-time) on 24 model, family and budget settings, with the number of points lost. (b) Points lost by each of the 21 strategies, averaged over the settings, at matched wall clock (left) and memory-time (right); the diamond marks the one strategy that loses nothing, a single BF16 attempt. (c) Least, median and largest losses of RetryPrec per budget and over both budgets, with the number of settings.

We add the switch to four existing systems, one per family (Table 34): a cascade, a self-repair loop (Olausson et al., 2024), adaptive stopping (Aggarwal et al., 2023) and speculative decoding. With both switched on, the solve rate equals the better of the two alone in all four families, so the two gains do not add up: the cascade, the self-repair loop and adaptive stopping already spend the budget on the problems that their first attempts leave open, the part of the budget that the switch would otherwise use for its second block. With the pair re-enumerated for the lower unit costs of speculative decoding, the two together reach 85.04 points on agent (Table 33), close to what additive gains would predict.

Table 34: Adding the switch to four existing systems, one per family, matched wall clock on Qwen2.5-32B. Solve rate of fixed L as the base, the system alone, RetryPrec alone and both together; the gain of each over the base schedule; the sum that additive gains would predict; and the residual, the combined gain minus that sum.

	Solve rate				Gain over base				
System (family)	base	system	RetryPrec	both	system	RetryPrec	both	sum	residual
Cascade (code)	72.12	75.55	75.55	75.55	+3.43	+3.43	+3.43	+6.86	−3.43
Self-repair (math)	76.30	77.50	76.72	77.50	+1.20	+0.42	+1.20	+1.62	−0.42
Adaptive stopping (reasoning)	78.71	80.42	81.87	81.87	+1.71	+3.16	+3.16	+4.87	−1.71
Speculative decoding (agent)	78.74	80.41	83.69	83.69	+1.67	+4.95	+4.95	+6.62	−1.67

G.5 Calibration

At matched wall clock, where every feasible pair has at most seven attempts of either of the two precisions, the schedule chosen from $k = 8$ attempts per calibration problem is the same as with $k = 64$ in each family, and as k varies from 8 to 128 only the math family moves, between $(2, 2)$ and $(4, 1)$; under memory-time the enumeration returns the pure low-precision schedule for all $k \geq 32$ (Table 32). With 64 calibration problems the reasoning family returns the pure low-precision schedule, while 128 problems come within 0.07 points of 512 problems in each of the four families (Table 12).

G.6 Failure Classes

Table 35 gives the numbers behind Fig. 3a. Compared with fixed L, RetryPrec shrinks the class of problems whose budget ran out by 3.28, 0.32, 3.04 and 4.80 points on code, math, reasoning and agent, and the class on which both precisions are near zero by 0.10 to 0.15 points in every family.

Table 35: Left: failure classes under fixed L and RetryPrec, share of the 240 problems of each family (in points) on which both precisions are near zero and whose budget ran out. **Right: the block length m chosen by the diagnostic variant** from eight attempts per problem against the enumerated best m in twelve model and recipe settings of the code family; they agree in 3 of the settings, and the mean absolute gap is 1.92 attempts.

Family	Schedule	near zero	ran out
Code	fixed L	8.02	19.85
	RetryPrec	7.88	16.57
Math	fixed L	7.15	16.55
	RetryPrec	7.05	16.23
Reasoning	fixed L	5.47	15.82
	RetryPrec	5.35	12.78
Agent	fixed L	5.42	15.84
	RetryPrec	5.27	11.04

Model / recipe	predicted	enumerated	gap
Qwen2.5-7B / GPTQ-W4	2	3	−1
Qwen2.5-7B / AWQ-W4	2	4	−2
Qwen2.5-7B / SmoothQuant-W8A8	2	3	−1
Qwen2.5-32B / GPTQ-W4	2	2	+0
Qwen2.5-32B / AWQ-W4	2	2	+0
Qwen2.5-32B / SmoothQuant-W8A8	6	1	+5
Qwen2.5-72B / GPTQ-W4	2	3	−1
Qwen2.5-72B / AWQ-W4	6	2	+4
Qwen2.5-72B / SmoothQuant-W8A8	2	4	−2
Mistral-24B / GPTQ-W4	2	6	−4
Mistral-24B / AWQ-W4	2	5	−3
Mistral-24B / SmoothQuant-W8A8	6	6	+0