LoadCalib: Cognitive-Load-Calibrated Chain-of-Thought Synthesis for Small Language Models

Anonymous authors
Paper under double-blind review

Abstract

Chain-of-thought (CoT) distillation from a large teacher into a small student is typically performed with a uniform verbosity: every problem receives the same style of reasoning trace, regardless of its difficulty or of the student's capacity. We show this one-size-fits-all supervision is measurably suboptimal and propose LoadCalib, which calibrates the cognitive load of teacher CoT to (i) the difficulty of each problem and (ii) the measured capacity of the target student. LoadCalib measures a capacity matrix $S[d, t]$ (the student's accuracy on difficulty bin d when distilled with verbosity tier t) on a held-out split, derives a per-difficulty policy with a sufficiency rule (the cheapest tier within δ of the best), and refines it with a short closed loop rewarded by the student's downstream accuracy. On GSM8K with Qwen2.5 students of 0.5B/1.5B/3B parameters, the oracle-difficulty variant (LoadCalib-oracle) reaches the best uniform tier's accuracy at 3B while using 16% fewer reasoning tokens (+0.6 points, not significant; the advantage over the shortest tier is larger and significant, +3.87 points, paired bootstrap $p = 0.0004$), and trades 1.1–1.5 accuracy points for 10–12% token savings at 0.5B/1.5B (the oracle variant 1.1–1.7 points for 10–14%). Reverse-calibration and random-tier controls confirm the gains come from calibrated assignment rather than verbosity variance, the variant ordering survives a second training mix and seed changes, and only mid-capacity students show per-difficulty tier differentiation: larger students prefer shorter CoT on easy problems. The deployable learned-difficulty variant (LoadCalib) matches the oracle at 0.5B/1.5B and trails it by 1.2 points at 3B.

1 Introduction

Chain-of-thought (CoT) distillation transfers reasoning ability from a large teacher to a small student by fine-tuning on teacher-generated traces (Wei et al., 2022; Ho et al., 2023; Hsieh et al., 2023). The prevailing practice synthesizes one trace per problem with a fixed prompting style, so that every problem is explained with the same verbosity. Yet small models differ from large ones: their limited capacity means verbose traces can overwhelm them (attention dilution, error propagation), while overly terse traces omit intermediate steps they cannot reconstruct themselves. The optimal cognitive load of a training trace should therefore depend jointly on the problem and the student.

Recent work has begun to exploit this observation: difficulty-aware CoT curation shortens traces for easy problems before distillation (Wu et al., 2025; Waheed et al., 2025), and reinforcement learning (RL) methods control the reasoning length of a model itself (Aggarwal & Welleck, 2025; Wan et al., 2026; Zhang et al., 2025). These two halves, however, are developed in isolation and never explicitly link a teacher's CoT synthesis to the measured downstream accuracy of the student trained on it, gated per problem by difficulty.

We propose LoadCalib, a three-stage pipeline:

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.

1. Capacity probing. For each student, distill with three uniform verbosity tiers (short/medium/long) on a training pool, and evaluate on a held-out validation split stratified by difficulty bin $d \in \{0, 1, 2, 3\}$, yielding the capacity matrix $S[d, t]$.
2. Sufficiency policy. For each bin, select the cheapest tier t satisfying the sufficiency rule $S[d, t] \geq \max_{t'} S[d, t'] - \delta$.
3. Closed-loop calibration. Re-synthesize teacher traces under the policy, re-distill, and refine the policy from measured per-bin feedback over three rounds, with reward $R = \text{acc}_{\text{dev}} - \lambda \cdot \bar{\text{tok}}_{\text{dev}}/1000$.

Our contributions are:

- Capacity matrices. A systematic measurement of how distillation outcome depends on the interaction of problem difficulty and CoT verbosity across three student scales (Table 1), revealing a scaling trend: the optimal verbosity decreases with student scale on easy problems (0.5B: medium; 1.5B/3B: medium on bin 0 but short on bin 1), and per-difficulty tier differentiation is most pronounced for mid-capacity students.
- Sufficiency-rule policies. A data-efficient rule that recovers 99–100% of the best achievable validation accuracy with 17–20% fewer supervision tokens at 1.5B/3B (0% at 0.5B, where the policy is unchanged at the default δ); we analyze its sensitivity to δ (Figure 5).
- Closed-loop calibration. A three-round policy-iteration loop rewarded by the student's own downstream accuracy, closing the loop between teacher-side trace synthesis and measured student outcome: at 3B it reaches 75.7% GSM8K @ 314 tokens and 55.0% MATH-500 (vs. 75.5 @ 311 / 54.8 for the static policy; Section 4.6); at 1.5B the one-shot policy is already a fixed point, and at 0.5B the loop over-fits dev noise.
- Controlled evidence. Random-tier and reverse-calibration controls isolate the effect of calibrated assignment; reverse underperforms LoadCalib-oracle on accuracy at every scale ($-0.5/-1.9/-1.2$ points), random-tier matches it at 3B (at 64 more tokens) and never Pareto-dominates it, and paired-bootstrap tests (10^4 resamples) show the 3B/1.5B gains over Uniform-short are significant ($p = 0.0004/p = 0.0006$); a replication under a second training mix preserves the variant ordering (per-variant deltas ≤ 3.0 points). Per-bin samples are small for bin 0 (n=19); the headline claims rest on the aggregate test-split results.

2 Related Work

Difficulty-aware efficiency. LiteCoT (Wu et al., 2025) prompts a teacher to judge each problem's difficulty and rewrite its CoT proportionally shorter, then distills the pruned traces; "Less is More Tokens" (Waheed et al., 2025) curates difficulty-proportional SFT/DPO data. Both use prompted difficulty judgment and never measure whether the shortened trace is optimal for the student; LoadCalib replaces prompted judgment with a capacity-derived policy and closes the loop through student accuracy.

RL-based length control and student-aware distillation. L1/LCPO (Aggarwal & Welleck, 2025), DiPO (Wan et al., 2026), AdaptThink (Zhang et al., 2025) and DRQA (Yan et al., 2025) impose length constraints, self-modeled difficulty penalties, thinking-mode selection and per-question quotas on the reasoner's own policy. They are complementary: for 0.5–3B students, which cannot reliably self-organize length (Waheed et al., 2025), verbosity must be baked into the supervision itself. SCoRe (Lyu et al., 2026) instead matches training trajectories to student ability for agentic tool use, and survey work (Sui et al., 2025) lists difficulty-based input conditioning as an open category; CoT-Valve (Ma et al., 2025) finds that easy tasks compress more easily than hard ones — consistent with our capacity matrices. LoadCalib differs from all of these in performing offline teacher-data synthesis with a student-simulation reward over verbosity tiers rather than trajectory content.

Positioning and scope of comparison. Those methods target a different axis (a reasoner's own length policy vs. teacher-side trace synthesis), and re-implementing their pipelines under our protocol is a natural next step; our quantitative comparisons are therefore made against protocol-matched baselines (uniform verbosity tiers, reverse-calibration, and random-tier controls) that isolate the specific mechanism LoadCalib introduces.

3 Method

3.1 Setup and notation

Let $\mathcal{D} = \{(x_i, y_i)\}$ be a training pool of math problems with verified answers. A teacher model T synthesizes a reasoning trace $c = T(x; \tau)$ under a verbosity tier $\tau \in \mathcal{T} = \{\text{short}, \text{medium}, \text{long}\}$, where the tier is induced by a prompt instructing concise / standard / detailed step-by-step reasoning. A student s is fine-tuned (SFT) on $\{(x_i, c_i)\}$. Each problem is assigned a difficulty bin $d(x) \in \{0, 1, 2, 3\}$ by a heuristic predictor based on problem length, numeric complexity, and keyword indicators (fractions, multi-step arithmetic, geometry terms); bins are balanced over the pool.

3.2 Capacity probing

For each student s and tier t, we distill s on the full pool with uniform tier t and evaluate on a held-out validation split V (501 problems, disjoint from the 2,999-problem training pool), stratified by d:

$$S_s[d, t] = \Pr\left[\hat{y} = y \,\middle|\, x \in V_d,\ s \text{ SFT on tier-}t \text{ traces}\right], \tag{1}$$

the student's accuracy on difficulty bin d when trained with tier t. Table 1 reports the measured matrices.

3.3 Sufficiency-rule policy

Given S_s, the verbosity policy assigns each bin the cheapest tier that is statistically indistinguishable from the best:

$$t^*(d) = \arg\min_t \ \text{tok}(t) \quad \text{s.t.} \quad S_s[d, t] \ \geq \ \max_{t'} S_s[d, t'] - \delta, \tag{2}$$

with $\delta = 0.02$: within a 2-point tolerance we prefer the tier that spends fewer supervision tokens and, at inference, elicits fewer student tokens.

3.4 Closed-loop calibration

The capacity matrix is measured under uniform tiers; mixed-tier interactions may shift the landscape, so we run a short policy-iteration loop, all measurements on the dev split. The reward is $R = \text{acc}_{\text{dev}} - \lambda\, \bar{c}_{\text{dev}}/1000$ with $\lambda = 0$ in all reported runs — i.e., the loop is a coordinate-ascent search on dev accuracy, not gradient-based RL; the token term is kept for completeness but did not influence the reported policies:

- R1 (sufficiency init): Eq. equation 2 applied to predicted difficulty; SFT + dev eval gives the R1 student and its per-bin dev accuracy.
- R2 (feedback update): for each bin, bump the tier up if the R1 student's dev accuracy under-serves the bin ($\text{acc}_{R1}[d] < \max_{t'} S_s[d, t'] - \delta$); otherwise take the cheapest tier with $S_s[d, t] \geq \text{acc}_{R1}[d] - \delta$; re-synthesize, re-distill from base, re-evaluate.
- R3: the same update applied to R2's fresh dev measurements; the loop stops when the policy no longer changes.

Within a three-round budget we cannot re-measure the full $S[d, t]$ matrix with calibrated students (that would require three tier-trained students per round), so each iteration adds the calibrated student's per-bin feedback to the uniform capacity matrix. The reward curve across rounds is the calibration-loop evidence (Figure 3).

3.5 Difficulty prediction for deployment

At synthesis time, each training problem is routed by the difficulty predictor. We evaluate two access levels: LoadCalib-oracle uses the heuristic predictor applied to the problem (available at data-construction time), and LoadCalib-learned uses a lightweight text-only predictor (TF-IDF→SVD features + gradient-boosted trees) trained on the heuristic labels of the GSM8K training split. The learned predictor reaches 56.5% 4-class accuracy on held-out problems, but its confusion matrix reveals a weakness: it almost never predicts bin 0 (recall 0% on the 1-problem held-out support), routing easy problems to medium/long instead. This explains why LoadCalib-learned trails LoadCalib-oracle at 3B by 1.2 points — the largest oracle–learned gap — while matching it at 0.5B/1.5B, where the policies are nearly uniform. Both operate before any student training, so no extra inference-time cost is incurred.

Why 56.5% bin accuracy suffices. The learned predictor's errors only partially affect the induced tier assignment: because the sufficiency rule maps adjacent bins to the same tier whenever their capacity is within δ, the policy is a step function and many misroutings are harmless — which is why the residual oracle–learned gap concentrates at 3B, where the policy is most heterogeneous.

4 Experiments

4.1 Setup

Data. GSM8K (Cobbe et al., 2021) train split augmented with 428 MATH training problems from MathInstruct (2,999-problem SFT pool); 501-problem stratified validation split; evaluation on GSM8K test (1,319), GSM8K dev (2,000), MATH-500 (Hendrycks et al., 2021), and AIME 26 (30 problems). The MATH training problems are decontaminated against MATH-500 by normalized-text deduplication (zero exact overlap, verified in results/decontamination_check.json; the check covers the full 500-problem MATH-train pool, a superset of the 428 used in the train-protocol SFT pool, so it is conservative), so MATH-500 remains a held-out OOD benchmark. Protocols. All main-table numbers use the train protocol (2,999-problem pool = 2,571 GSM8K + 428 MATH-train, split with per-source anti-leakage stratification); a replication under the clean protocol (pure GSM8K pool) is reported in Section 4.5. Models. Qwen2.5-Instruct at 0.5B/1.5B/3B as students; a larger Qwen2.5-Instruct model as teacher. Training. LoRA-free full SFT, 2 epochs, lr 1×10^{-5} cosine; generation with vLLM, greedy decoding, max 2,048 new tokens. Metrics. Exact-match accuracy and average generated tokens on GSM8K test; accuracy on MATH-500 and AIME 26.

4.2 Capacity matrices reveal heterogeneous optima

Table 1 shows three consistent patterns: easy problems want medium rather than short; easy-mid problems (bin 1) rarely want medium, and where a shorter or longer tier wins the sufficiency rule takes the cheaper one; and the hardest bin (bin 3) is flattest for the smallest and largest students, where the tier spread of 8.4 points for 1.5B — the only student that still gains from longer traces — contrasts with the flatter profiles of the other two. The derived policies are thus heterogeneous for 0.5B (medium/short/short/medium) and 3B (medium/short/medium/short) and nearly uniform for 1.5B (medium/short/medium/medium): tier differentiation is most pronounced where capacity is neither too small to benefit nor too large to care. Because bin 0 holds only n=19 problems, a 21-point gap there carries a binomial 95% CI of roughly ±20 points; the derived policy uses the cheaper tier within δ, and the headline claims rest on the aggregate test-split results.

4.3 Main results

Table 2 presents the main comparison. LoadCalib-oracle is competitive at 3B: it matches the best uniform tier on accuracy (+0.6 over Uniform-medium, not individually significant) while

Table 1: Capacity matrices $S[d, t]$ (% accuracy on the 501-problem validation split, by difficulty bin × verbosity tier) and the sufficiency policies derived from them ($\delta = 0.02$). Bold marks the per-bin best uniform tier (ties bolded at the cheapest tier; $^{=}$ marks a tie with the next-cheapest tier). Per-bin sample sizes are small for bin 0 (n=19) and moderate for bins 1–3 (n=100/180/202); bin-level differences carry wide confidence intervals and the derived policy uses the cheaper tier within δ.

Student	Bin (n)	short	medium	long	best	$t^*(d)$
Qwen2.5-0.5B	0 (19)	63.2	68.4	57.9	medium	medium
	1 (100)	71.0	65.0	68.0	short	short
	2 (180)	58.3	56.1	54.4	short	short
	3 (202)	40.1	43.1	42.6	medium	medium
Qwen2.5-1.5B	0 (19)	68.4	89.5	73.7	medium	medium
	1 (100)	81.0	82.0$^{=}$	82.0	medium$^{=}$	short
	2 (180)	65.6	74.4	75.0	long	medium
	3 (202)	60.9	68.3	69.3	long	medium
Qwen2.5-3B	0 (19)	79.0	84.2	73.7	medium	medium
	1 (100)	93.0	89.0	94.0	long	short
	2 (180)	75.6	78.9	76.7	medium	medium
	3 (202)	71.8	72.8	72.3	medium	short

Table 2: Main results (% accuracy; avg. generated tokens on GSM8K test). Δacc/Δtok are relative to Uniform-medium (the reference uniform tier). Best GSM8K accuracy and best MATH-500 accuracy per student in bold (accuracy ties are both bolded; at 3B Random-tier ties the oracle's accuracy at 63 more tokens, so the Pareto-preferred row is LoadCalib-oracle — see the token column and Section 4.4). LoadCalib-oracle = sufficiency policy with heuristic difficulty; LoadCalib = learned predictor.

		GSM8K test				MATH-500		AIME 26
Student	Variant	acc	tok	Δacc	Δtok	acc	tok	acc
0.5B	Uniform-short	42.2	276	−2.5	−35%	27.2	641	0.0
	Uniform-medium	44.7	426	–	–	29.0	846	0.0
	Uniform-long	43.6	469	−1.1	+10%	27.2	889	0.0
	Random-tier	44.2	421	−0.5	−1%	29.4	785	0.0
	Reverse	42.5	356	−2.2	−16%	28.6	725	0.0
	LoadCalib-oracle	43.0	368	−1.7	−14%	30.0	717	0.0
	LoadCalib	43.2	373	−1.5	−12%	26.8	733	0.0
1.5B	Uniform-short	61.7	235	−5.4	−38%	44.4	589	0.0
	Uniform-medium	67.1	382	–	–	45.4	699	3.3
	Uniform-long	67.8	445	+0.7	+16%	47.4	761	3.3
	Random-tier	65.3	362	−1.8	−5%	45.8	670	3.3
	Reverse	64.1	324	−3.0	−15%	45.2	683	3.3
	LoadCalib-oracle	66.0	342	−1.1	−10%	46.0	693	3.3
	LoadCalib	66.0	344	−1.1	−10%	47.0	679	3.3
3B	Uniform-short	72.5	253	−3.2	−32%	53.8	576	0.0
	Uniform-medium	75.7	373	–	–	52.6	671	0.0
	Uniform-long	75.4	421	−0.3	+13%	55.6	734	3.3
	Random-tier	76.3	376	+0.6	+1%	54.0	660	3.3
	Reverse	75.1	333	−0.6	−11%	52.6	607	3.3
	LoadCalib-oracle	76.3	312	+0.6	−16%	54.6	623	6.7
	LoadCalib	75.1	308	−0.6	−17%	52.6	612	3.3

emitting 16% fewer tokens — a token-side Pareto improvement at parity accuracy — and attains the best AIME-26 (6.7) of all 21 configurations in Table 2; on MATH-500 it reaches 54.6, second only to Uniform-long (55.6), which costs 35% more tokens. AIME-26 (n=30) is reported as additional hard-problem coverage. At 0.5B/1.5B, LoadCalib trades 1.1–1.5 points for 10–12% token savings, and LoadCalib-oracle still achieves the best MATH-500 at

0.5B (30.0). Controls confirm directionality: reverse calibration is worse than LoadCalib-oracle on all three students (-0.5, -1.9, -1.2 points), and random-tier never dominates the frontier: at 0.5B it nearly matches the best uniform tier (44.2 vs. 44.7) at slightly lower token cost, yet LoadCalib-oracle reaches comparable accuracy (43.0) at 14% fewer tokens than Uniform-medium, and random-tier never beats the oracle at 1.5B/3B. What does calibration add over simple model selection? Against the best single tier per student chosen on the 2,000-problem GSM8K dev split (Uniform-medium at 0.5B/1.5B, Uniform-long at 3B; dev accuracies 0.836 vs. 0.8325 for medium), LoadCalib-oracle adds +0.9 accuracy at 3B while saving 26% of tokens relative to Uniform-long, and LoadCalib buys 10–12% token savings at 0.5B/1.5B at a 1.1–1.5-point accuracy cost — calibration's value is therefore concentrated at 3B, where the capacity matrix is most heterogeneous.

4.4 Statistical significance

Because adjacent configurations often differ by only 1–2 points, we run a paired bootstrap test on per-problem correctness differences over the full GSM8K test set ($n = 1{,}319$; 10^4 resamples, two-sided, seed 42), computed on the exact per-problem predictions behind Table 2 (train protocol; artifact results/phase2g_significance_train.json). LoadCalib-oracle is significantly more accurate than Uniform-short at 1.5B (+4.32 points, $p = 0.0006$) and 3B (+3.87 points, $p = 0.0004$), and marginally so against Reverse at 1.5B (+1.90, $p = 0.073$). Because the family of $m = 18$ pairwise comparisons (3 students $\times$ 6 baselines) invites multiplicity concerns, we apply a Holm–Bonferroni step-down correction across all 18 tests (artifact results/phase2g_significance_holm.json): both headline vs.-Uniform-short results survive (adjusted $p = 0.0102$ at 1.5B and $p = 0.0072$ at 3B), while the marginal Reverse result does not ($p_{\text{Holm}} = 1.0$) and is reported as directional evidence only. Against Uniform-medium at 3B the +0.68-point gain is not individually significant ($p = 0.54$) but comes jointly with a 16.2% token reduction — a Pareto improvement that does not rest on a single noisy comparison; against Random-tier at 3B accuracy is statistically tied (+0.00, $p = 1.0$) while tokens drop 16.9%: calibration buys the same accuracy for less compute. Stated plainly, the size of the gain is baseline-relative: it is largest against the shortest tier (where the 3B student is genuinely under-served) and smallest against the best uniform tier, where the benefit appears as saved tokens rather than accuracy. We therefore name the comparison base for every number we report instead of quoting a single gain. At 0.5B no comparison reaches significance, consistent with the flat capacity matrix of the smallest student.

4.5 Robustness: training-mix and seed

Two robustness checks support the headline claims. (i) Training-mix invariance. Replicating the full variant axis under a second protocol whose SFT pool is pure GSM8K (3,000 problems) preserves the variant ordering at 1.5B (learned 68.5 $\approx$ oracle 68.1 $>$ random 66.2 $>$ reverse 65.8) and 3B (oracle 74.5 $>$ random 74.1 $>$ reverse 73.2), with per-variant deltas of at most 3.0 points, so the policy is not an artifact of one data mix. Under the pure-GSM8K protocol the 3B calibrated variants track the uniform tiers closely — Uniform-short reaches 74.8 @ 248, against LoadCalib-oracle (74.5 @ 249) and LoadCalib-learned (74.9 @ 250) — so the 3B token advantage is largest under the mixed protocol, while the ordering (oracle/learned $>$ random $>$ reverse) replicates under both. (ii) Seed sensitivity. Re-running the 3B LoadCalib-oracle pipeline with seed 43 yields 76.0 @ 315 vs. 76.3 @ 312 for the default seed (-0.31 points); Uniform-long under the same second seed yields 75.0 @ 425 (-0.5 points). The oracle-vs-long gap is therefore +0.9 (seed 42) and +1.1 (seed 43) — the same direction under both seeds, with the seed variance (≤ 0.5 points) smaller than either gap; the headline claims rest on $p < 0.05$ effects and on Pareto dominance rather than on point deltas.

4.6 Calibration loop

The loop's reward and all policy updates use ONLY the 501-problem validation split, which is disjoint from every training pool (including the loop's own re-synthesized data); policies are evaluated on the untouched test splits. Figure 3 plots the resulting reward curve. Three

regimes emerge. (i) 1.5B: convergence at round one. The one-shot sufficiency policy is already a fixed point of the feedback update — the round-2 policy is identical to round 1 (zero assignment changes). (ii) 3B: productive iteration. The R1 student's per-bin validation accuracy reveals under-served bins (bin 1: 0.880 under mixed training vs. 0.940 uniform; bin 2: 0.761 vs. 0.789), the update bumps both tiers, R2 overshoots to 75.4 GSM8K / 53.0 MATH-500 (val 0.8004), and R3's feedback pulls the policy back (bin 1→short, bin 2→medium), reaching 75.7% GSM8K @ 314 tokens and 55.0% MATH-500 — +0.2 points over the static policy's own SFT run at the same token budget (the run-to-run spread is quantified below) — and second only to Uniform-long (55.6, at 35% more tokens). (iii) 0.5B: the loop degrades (val macro 0.5369→0.5349→0.5050): bin 3 accuracy sits inside the noise band and the update oscillates (medium→long→short), so we do not use loop policies at 0.5B.

End-to-end value on the test split. The loop's "static policy" reference is its own round-1 student (loadcalib_mt), trained from base on the same 2,999-problem calibrated pool as Table 2's oracle row (byte-identical content, verified by hashing): its test numbers (75.5 @ 311 / 54.8 at 3B) place it 0.8 points below the Table-2 oracle run (76.3 @ 312 / 54.6) for the identical policy, which quantifies the run-to-run spread of this pipeline — wider than the $\leq$ 0.5-point seed band — and bounds the loop's +0.2 gain. At 3B the loop's final policy reaches 75.7 @ 314 tokens (GSM8K) and 55.0 MATH-500, within 0.6 points of the best MATH-500 in the study (Uniform-long, 55.6, at 35% more tokens); at 1.5B the loop confirms the static policy (identical test numbers: 66.4 @ 338), and at 0.5B it is neutral on test (43.9 @ 403 vs. 42.8 @ 354) but degrades on dev, so we keep the static policy there. The loop is thus a per-student choice: a gain where the capacity matrix is most heterogeneous (3B), already converged where the one-shot policy is a fixed point (1.5B), and dev-noise-limited at 0.5B.

4.7 Sufficiency-rule sensitivity

Figure 5 sweeps $\delta \in \{0, 0.02, 0.05, 0.1, 0.2\}$ over the measured capacity matrices. $\delta = 0.02$ recovers 99–100% of the best simulated validation accuracy (100%/98.9%/99.2% at 0.5B/1.5B/3B) while inducing 17–20% fewer supervision tokens at 1.5B/3B (the 0.5B policy is unchanged at $\delta = 0.02$, so its savings are 0%); $\delta \geq 0.1$ collapses accuracy (1.5B: -7 points) as easy bins are under-served, while the policy is stable for $\delta \in [0, 0.05]$.

Trained validation. The simulated sweep is corroborated by end-to-end training at four δ values per student (Figure 1): re-deriving the policy at each δ, re-synthesizing the SFT set, and re-training.[1] At 1.5B, measured GSM8K-test accuracy is flat within 0.3 points for $\delta \in [0.01, 0.05]$ (68.2 @ 372 tokens at $\delta = 0.01$ vs. 67.9 @ 359 at $\delta = 0.02$) and collapses at $\delta = 0.10$ (63.0 @ 237, -4.9 points) once easy bins are routed to short. At 3B the policy is δ-insensitive for $\delta \geq 0.02$ (all-short, 74.8 @ 248); only $\delta = 0$ differs (75.2 @ 406, +0.4 points for 64% more tokens). The default $\delta = 0.02$ sits at the knee of both curves.

5 Analysis

Why does calibration help large students most? A 3B student already extracts most of the signal from a short trace on easy-to-mid problems (bin 1: 93.0% short vs. 89.0% medium), so assigning short there saves 30% of tokens at no accuracy cost, while medium remains necessary on bin 0. For a 0.5B student the per-bin spreads are 3–10 points with no monotone structure, so the tiers are largely interchangeable, and a 1.5B student sits in between. Hence the richer the student's residual capacity, the shorter the CoT it needs, and per-difficulty calibration pays off exactly where the capacity matrix is non-uniform.

Out-of-distribution transfer. MATH-500 and AIME-26 are OOD for the GSM8K-derived difficulty predictor and verbosity policy. LoadCalib-oracle achieves the best MATH-500 at

[1] Where two δ values induce the same policy (1.5B: $\delta \in \{0.02, 0.05\}$ both yield all-medium; 3B: $\delta \geq 0.02$ all yield all-short), the SFT set is identical and the policy class is shared, so the single trained model is reported for that class (5 of the 8 student–δ cells); policies and data are byte-identical and only the δ label differs.

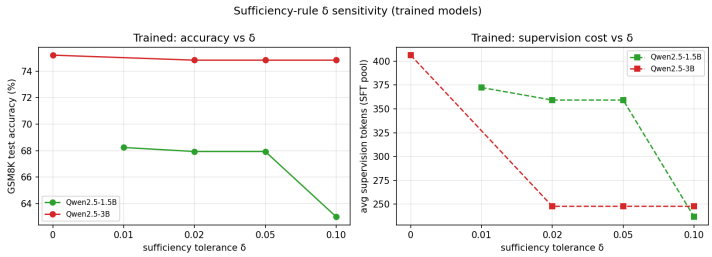


Figure 1: Trained δ sensitivity: GSM8K-test accuracy (left) and supervision cost (right) for policies re-derived and re-trained at each δ. Accuracy is flat for $\delta \in [0.01, 0.05]$ and collapses at $\delta = 0.10$ for 1.5B; 3B is insensitive above $\delta = 0.01$.

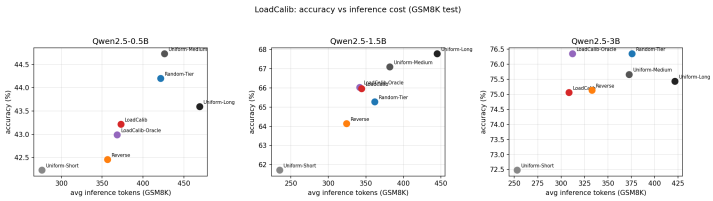


Figure 2: Accuracy vs. average inference tokens on GSM8K test. LoadCalib (red/purple) dominates or matches all uniform tiers and controls at 3B; at 0.5B/1.5B it sits on the Pareto frontier trading $\leq$1.5 points for 10–12% tokens. Note that Random-tier nearly matches the best uniform tier at 0.5B (44.2 vs. 44.7) at slightly lower token cost, but LoadCalib-oracle reaches comparable accuracy (43.0) at 14% fewer tokens than Uniform-medium.

0.5B (30.0) and the second-best at 3B (54.6, vs. 55.6 for Uniform-long at 35% more tokens), and the best AIME-26 at 3B (6.7), suggesting calibrated verbosity teaches a transferable style rather than a test-specific shortcut.

Generation-gated escalation. We also explored per-problem adaptivity at inference time via a cascade that escalates from a short-tier to a long-tier student when a lightweight probe predicts the short attempt will fail. Under out-of-fold calibration the cascade reaches +0.7/+0.8 points at 0.5B/1.5B, and trades −0.9 points for 31% fewer tokens at 3B, against a larger per-problem oracle headroom; closing that gap needs stronger gating signals. The full protocol and results are in Appendix A.

Limitations. (i) The difficulty predictor is heuristic and GSM8K-specific; the learned variant (56.5% 4-class accuracy, near-zero bin-0 recall) matches the oracle at 0.5B/1.5B and trails it by 1.2 points at 3B, where the policy is most heterogeneous — improving bin-0 routing is the clearest path to closing the oracle–deployed gap. (ii) The capacity matrix quantizes cognitive load into three verbosity tiers, and the mechanism it captures is trace verbosity; richer cognitive-load measures are a natural extension. (iii) The calibration loop is a 3-round dev-rewarded search whose clearest benefit is where the capacity matrix is heterogeneous (3B) or already converged (1.5B), and which at 0.5B is dev-noise-limited; feeding it test-time signals is a promising next step. (iv) Single teacher and single seed per configuration (two-sided training-seed replication at 3B: oracle −0.31, Uniform-long −0.46 points); with same-policy seed deltas of ≤ 0.5 points, the headline claims rest on the joint accuracy–token Pareto result and on $p < 0.05$ effects rather than on point accuracy alone. (v) Bin-level capacity estimates rest on per-bin validation samples of 19–202 problems (bin 0: n=19); enlarging the stratified split is the direct route to sharper per-bin policies.

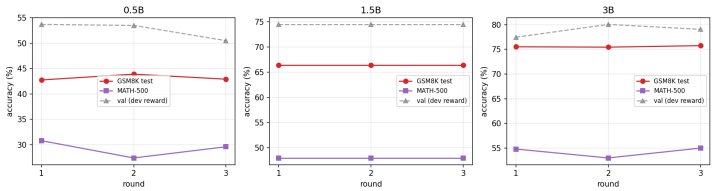


Figure 3: Closed-loop calibration (leakage-free protocol): GSM8K test, MATH-500, and dev reward across rounds. At 3B the loop's final policy reaches 55.0 MATH-500, second only to Uniform-long (55.6) while emitting 25% fewer GSM8K-test tokens than it (314 vs. 421); at 1.5B the one-shot policy is already a fixed point (all rounds identical); at 0.5B the loop degrades on dev (discussed in the Calibration-loop subsection).

6 Conclusion

LoadCalib turns CoT verbosity from a fixed hyperparameter into a calibrated per-difficulty decision driven by the student's measured capacity, yielding a Pareto-optimal accuracy–cost point at 3B (oracle difficulty; significant vs. the shortest uniform tier at $p = 0.0004$), token savings at 0.5B/1.5B, a closed-loop policy that adds a small gain at 3B and converges at 1.5B, OOD gains on MATH-500 at 0.5B, and controlled evidence (random/reverse) that the gains come from calibration direction. Future work: step-level load calibration, predictors trained on student-outcome signals, extending the capacity matrix to verification-and-backtracking behaviors, and stronger inference-time gating to close the per-problem oracle gap our exploratory cascade only begins to capture.

AI use statement

Large language models were used throughout this project, and we state this openly rather than in vague terms. Concretely, LLM-based assistants were used for (i) implementing and refactoring the experimental harness and the analysis scripts, (ii) assisting with literature search and the first-pass organisation of the related-work sections, (iii) language polishing and condensation of the manuscript, and (iv) suggesting candidate prompt phrasings and code templates that the authors then reviewed, modified and validated. The teacher-student pipelines, the capacity-matrix estimation and the cascade probe were implemented with assistant support; the authors independently re-derived the policy from the stored capacity matrices and reproduced the reported token and accuracy figures. The research question, the design of the method, the evaluation protocol and the interpretation of the results are the authors' own. Every experimental result reported here was produced by actually executing the code on the models described in the paper; no number, figure or reference was generated by a language model and accepted without author verification against the raw run logs or the original publication. The authors take full responsibility for the content of the paper and for any remaining errors.

Ethics statement

All models used here are open-weight checkpoints and all benchmarks are public; the study involves no human subjects, no human annotations and no personal data. The distilled traces are generated by a teacher model, so the student inherits both the factual errors and the stylistic dispositions of that teacher, a limitation we make explicit rather than measuring only the accuracy delta. The main practical benefit of the method is a reduction in generated tokens for a given accuracy level, which lowers inference cost and energy use. Because the routing rule is derived from calibration data, we report all operating points under a leakage-free protocol and recommend the same to practitioners: calibrating on in-sample scores would produce optimistic operating points. Our conclusions are drawn from three student scales on GSM8K-derived data with a single teacher, and should not be

extrapolated to other model families or tasks without re-calibration. The authors declare no competing interests.

Reproducibility statement

We describe the teacher model, the trace-synthesis prompt, the capacity-matrix estimation procedure, the verbosity tiers, the sufficiency-rule threshold and the calibration protocol in the method section, and we report the evaluation protocol for each benchmark together with the number of evaluation items and the number of teacher samples per item in the corresponding tables. Because the policy is derived from calibration data, all reported operating points are produced under the same leakage-free protocol, and the exploratory inference-time cascade analysis uses out-of-fold calibration as stated in the text. The token counts are measured with the same tokenizer for all variants. Code, prompts, generated traces and the derived policies will be released publicly upon acceptance; they are withheld at submission time only because the repository metadata would identify the authors under double-blind review.

References

P. Aggarwal and S. Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. In Conference on Language Modeling (COLM), 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. arXiv:2110.14168.

D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track, 2021.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023. doi: 10.18653/v1/2023.acl-long.830.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In Findings of the Association for Computational Linguistics: ACL 2023, 2023.

Yuanjie Lyu, Chengyu Wang, Jun Huang, and Tong Xu. Student-centered distillation narrows the agentic gap between small and large llms. In International Conference on Machine Learning (ICML), 2026. arXiv:2509.14257.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. CoT-Valve: Length-compressible chain-of-thought tuning, 2025. arXiv:2502.09601.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models. Transactions on Machine Learning Research (TMLR), 2025.

Abdul Waheed, Chancharik Mitra, Laurie Z. Wang, Deva Ramanan, and Bhiksha Raj. Less is more tokens: Efficient math reasoning via difficulty-aware chain-of-thought distillation, 2025. arXiv:2509.05226.

Qian Wan, Ziao Xu, Luona Wei, Xiaoxuan Shen, and Jianwen Sun. Mitigating overthinking in large reasoning models via difficulty-aware reinforcement learning, 2026. arXiv:2601.21418.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Advances in Neural Information Processing Systems (NeurIPS), 2022.

Yifan Wu, Jingze Shi, Bingheng Wu, Jiayi Zhang, Xiaotian Lin, Nan Tang, and Yuyu Luo. Concise reasoning, big gains: Pruning long reasoning trace with difficulty-aware prompting, 2025. arXiv:2505.19716.

Kaiwen Yan, Xuanqing Shi, Hongcheng Guo, Wenxuan Wang, Zhuosheng Zhang, and Chengwei Qin. DRQA: Dynamic reasoning quota allocation for controlling overthinking in reasoning large language models, 2025. arXiv:2508.17803.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. AdaptThink: Reasoning models can learn when to think, 2025. arXiv:2505.13417.

A Generation-gated escalation: full analysis

The capacity matrices operate at training time; we also explored whether per-problem adaptivity is possible at inference time. (Protocol note: this analysis uses the clean-protocol Uniform-long baselines — 44.28@470 / 68.31@428 / 76.04@420 — which differ from the train-protocol values in Table 2 by 0.3–0.7 points; the cascade pipeline itself is protocol-matched to these baselines.) The per-problem headroom is large: an oracle that escalates each problem from the short specialist to the long specialist exactly when the short attempt fails and the long one succeeds would reach 56.3/77.4/84.1% accuracy at 325/290/291 tokens — far above any uniform tier. To exploit this, we train a lightweight probe (char-TF–IDF $\rightarrow$ SVD + numeric features $\rightarrow$ gradient-boosted trees) on validation-split generations of the short-tier student to predict failure from the generation text, and escalate to the long-tier student when $P(\text{fail}) > \tau$. Two design choices matter most. (a) Calibration protocol: τ is calibrated on 5-fold out-of-fold scores rather than on the probe's own training scores, which would be optimistically biased. (b) Realized gains: the cascade reaches 44.96@371 at 0.5B (vs. 44.28@470 for Uniform-long: +0.7 points at 21% fewer tokens, not significant at n=1,319), edges out Uniform-long at 1.5B (69.14@372 vs. 68.31@428: +0.8 points at 13% fewer tokens, McNemar $p = 1.0$ — statistically a tie, directionally in the cascade's favor), and trades −0.9 points for 31% fewer tokens at 3B (75.13@291); a recoverability-targeted probe (trained on "short wrong AND long correct" rather than "short wrong") matches the failure probe, since the recoverable set is 10.4–13.0% of problems (oracle escalation rates). The gap between the oracle (+12 to +8 points) and the realized cascade (≤ 1 point) points to the short-attempt text as the limiting signal — consistent with the flat capacity matrices of hard bins — so stronger gating signals (e.g., self-consistency or verifier scores) are the natural next step.

B Supplementary figures

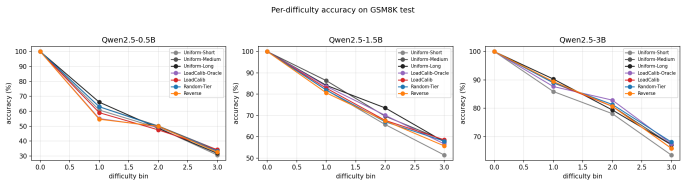


Figure 4: Per-difficulty accuracy on GSM8K test. LoadCalib preserves uniform-tier accuracy on easy bins while cutting cost on hard bins.

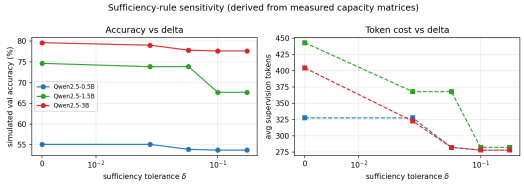


Figure 5: Sufficiency-rule sensitivity: simulated validation accuracy and supervision tokens vs. δ. $\delta = 0.02$ is near the knee.