WHAT DO ROLLOUT TOKENS BUY? HIDDEN COSTS AND BUDGETED ACCURACY IN RLVR

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) pays for every token it generates, yet training choices are rarely judged by what those tokens buy. We study this question on competition mathematics with two training choices and one evaluation choice. First, we replace one third of a SmolLM3-3B training set with harder problems and keep the other 2,048 problems at the same positions. Responses to these unchanged problems become twice as long (+101%) at a similar training reward, and late in training most of their tokens are in responses that reach the length limit. Second, we halve the training response limit of Qwen3.5-9B from 4,096 to 2,048 tokens. Training compute falls by 28%, and majority-vote accuracy under a 16,384-token budget rises from 36.1% to 43.5% as far more evaluation answers finish. The gain remains when truncated answers are continued to 8,192 tokens. Third, under the same budget, counting problems with any correct answer and counting the answer a model returns can rank a base model and its RL-trained version in opposite orders. The SmolLM3-3B base finds a correct answer on more of 216 problems (57 against 39) but returns a correct majority answer on fewer (5 against 18), and majority vote favors the trained model in all three model pairs. In all three cases, more tokens did not mean more correct answers. We therefore suggest judging an RLVR training choice by the tokens it adds on problems that stay the same and by the answers the trained model returns under a fixed token budget.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) trains a language model by sampling several responses to each problem, checking whether their final answers are correct, and updating the model

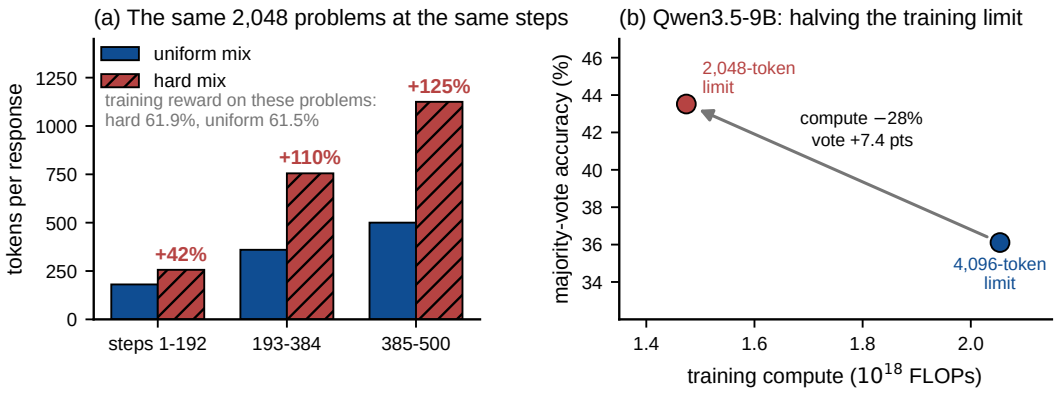


Figure 1: **Two training choices and what their tokens buy.** (a) Replacing one third of a SmolLM3-3B training set with harder problems makes the *unchanged* 2,048 problems more expensive in every pass through the training set, at a similar training reward (Section 4). (b) Halving the training response limit of Qwen3.5-9B lowers the training compute and raises majority-vote accuracy at 16,384 tokens per problem (Section 5).

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.

with those rewards (Shao et al., 2024). Every RLVR run sets a maximum response length. A response that reaches this limit is cut off and earns a reward only if its answer appears before the cut. A higher limit leaves room for long reasoning, but long responses cost tokens, and in synchronous training a few of them can stall a whole generation batch (Zhou et al., 2025). If a run uses a 4,096-token limit, does it need 4,096 tokens?

The answer depends on why responses become long. A problem may require a long derivation, or the model may reach the answer by trial and error, after approaches that fail. Competition mathematics such as the AIME (Mathematical Association of America, 2026) and HMMT (HMMT, 2026) is designed to be solved by hand under time pressure: the AIME gives three hours for 15 problems, and each answer is an integer from 0 to 999 (Mathematical Association of America, 2026). A short solution therefore usually exists, and a long response more likely contains failed attempts. In GRPO, every token of a response receives the response's advantage, so a correct response that succeeds after failed attempts also reinforces those attempts, and with token-level loss aggregation (Yu et al., 2025) a response twice as long contributes twice as many positively weighted tokens. This may teach the model to explore in context (Setlur et al., 2025), or it may teach long responses even for problems with short solutions (Liu et al., 2025; Hou et al., 2026; Bounhar et al., 2025). A training limit is the simplest way to favor shorter responses, because only answers reached within the limit can earn a reward.

The choice of training problems raises a related question. Hard problems elicit long responses, and many methods select training problems by difficulty (Section 2). A harder training set can raise the average response length because the added problems need longer responses, or because training on them lengthens the responses to all other problems. An average over the dataset cannot separate the two, and only the second adds cost to problems that were in the dataset all along.

We study both questions on competition mathematics. To isolate the effect of difficulty, we train on two training sets that share two thirds of their problems at the same positions in the training order. To test the limit, we halve it from 4,096 to 2,048 tokens. Because both interventions change response length, a fixed number of attempts per problem would give more inference compute to the model that writes longer responses (Brown et al., 2024). We therefore give every model the same number of generated tokens per problem and report *majority-vote accuracy*, the accuracy of the most frequent answer among the responses that finish within this budget (Wang et al., 2024). Finally, we ask whether counting any correct answer and counting the returned answer rank models in the same order under one budget.

Our three findings are as follows.

Harder problems raise the cost of unchanged problems. On SmolLM3-3B, the shared problems cost 101% more tokens under the harder training set, at a similar training reward. The increase starts as longer finished responses and grows into responses that run to the limit (Section 4).

A shorter training limit lowers cost and improves budgeted accuracy. On Qwen3.5-9B, halving the limit cuts the training compute by 28% and raises majority-vote accuracy by 7.4 points. More of the model's answers finish: 48% of its evaluation responses end within the evaluation limit, against 30% for the 4,096-token model. The gain remains under a longer evaluation limit, and the coverage advantage remains at an equal number of attempts (Section 5).

Coverage and majority vote can give opposite rankings under the same budget. A SmolLM3-3B base model has a correct answer on more problems than its GRPO-trained version, 57 against 39, but a correct majority answer on fewer, 5 against 18. Majority vote favors the trained model in all three model pairs, and so does requiring a supported derivation (Section 6).

Together, these results show that more rollout tokens can mean a hidden cost on other problems, a training limit whose extra length goes to unfinished answers, or many cheap attempts whose correct answers the model does not return. Appendix A maps each claim to its evidence. The experiments use base models from 3B to 35B parameters and mathematics problems with binary rewards.

2 Related work

Problem difficulty and data selection in RLVR. RLVR trains on automatically checked answers (Lambert et al., 2024; Guo et al., 2025), and in GRPO a group whose rewards all agree has zero

advantage (Shao et al., 2024). Many methods therefore choose training problems by difficulty. Some sample problems with intermediate success rates (Foster et al., 2025; Bae et al., 2025), DAPO filters groups whose rewards all agree (Yu et al., 2025), selective rollouts skip prompts that are unlikely to be informative (Zheng et al., 2025), and other methods allocate more rollouts to hard problems (Li et al., 2025; Noukhovitch et al., 2026). Which problems help most is unsettled: Cheng et al. (2026) and Bao et al. (2026) report the largest gains from easy and medium problems, and Pikus et al. (2025) from the hardest ten percent. Closest to our shared-problem result, prior work links unsolved problems to verbosity and finds that moderately easy problems act as a length regularizer for the whole dataset (Fatemi et al., 2025; Bounhar et al., 2025). Our design instead holds the measured problems and their positions fixed, so the added cost is a change in how the model answers them.

Controlling response length. GRPO's normalization can bias response length (Liu et al., 2025). ThinkPrune trains under progressively tighter length limits, L1 trains the model to follow a requested length, and difficulty-aware training cuts tokens on easy problems (Hou et al., 2026; Aggarwal & Welleck, 2025; Chen et al., 2025). DAPO reshapes the reward of overlong responses (Yu et al., 2025), e3 pairs harder problems with larger training budgets to teach in-context exploration (Setlur et al., 2025), and Wachi et al. (2026) relate training gains to the generation limit relative to the tokens needed for a first correct solution. We instead measure the simplest intervention, a shorter training limit, by its training compute and by the answers returned under a token budget.

Budget-aware evaluation, coverage, and reasoning quality. Repeated sampling raises coverage with the number of attempts (Brown et al., 2024). Base models can overtake RL-trained models at large attempt counts (Yue et al., 2025), although a statistical test finds no established crossover in five public model pairs (Yang et al., 2026), and GRPO can lower coverage at large attempt counts by making rare correct responses less likely (He et al., 2025; Zhou, 2026). Rankings of models can depend on the inference budget (Wang et al., 2024; Guedes de Souza & Panisson, 2026), and success within a fixed number of generated tokens has been used as a metric (Brandfonbrener et al., 2024). CoT-Pass@k requires both a correct answer and correct reasoning (Wen et al., 2025), and FAPO targets responses with a correct answer but flawed reasoning during training (Ding et al., 2026). We add a case in which, under one budget, budgeted coverage and majority vote rank a base model and its trained version in opposite orders.

3 SETUP

Models and training. We use three base models, SmolLM3-3B, Qwen3.5-9B, and Qwen3.5-35B-A3B, and train them with GRPO and a binary answer reward (Shao et al., 2024). Each update samples eight responses for each of 16 problems. For response i in a group of $G = 8$ responses to the same problem, the advantage is

$$A_i = r_i - \frac{1}{G}\sum_{j=1}^{G} r_j, \qquad r_i \in \{0, 1\}. \tag{1}$$

There is no division by the group's reward standard deviation and no KL penalty. The loss averages over all response tokens in the update, so every token of response i carries the coefficient A_i/T, where T is the number of response tokens in the update. If one of eight responses is correct, its advantage is $7/8$ and each incorrect response has $-1/8$. If all eight rewards agree, every advantage is zero. SmolLM3-3B is trained with full parameters in verl for 500 updates (learning rate 10^{-6}). Qwen3.5-9B and Qwen3.5-35B-A3B are trained with rank-32 LoRA for 192 updates on 3,072 problems (learning rate 10^{-5}). The training response limit is 4,096 tokens unless stated otherwise. Training problems come from GSM8K, MATH, and competition-style sources, and "hard" problems are those with a low success rate for the base model before training. The two model families differ in update method and backend, so we do not compare across them (Appendix B).

Experiments. Section 4 trains SmolLM3-3B on two training sets that share 2,048 of their 3,072 problems. Section 5 trains Qwen3.5-9B with a 4,096- and a 2,048-token training limit. Section 6 compares each base model with its GRPO-trained version from Table 1. We estimate training compute from token counts as $2N(T_{\text{in}} + T_{\text{gen}}) + 6N\,T_{\text{train}}$, where N is the number of parameters and T_{in}, T_{gen}, and T_{train} are the uncached input, generated, and training-pass tokens (Appendix E).

Table 1: **Where the training tokens go in the three GRPO runs.** "At limit" marks a response that reaches the 4,096-token training limit, and the last column gives the share of those responses that receive a reward. The two categories overlap. SmolLM3-3B counts response-mask positions over 500 updates, and the Qwen runs count generated tokens over 192 updates.

Model	Response tokens (M)	Tokens in zero-advantage groups	Responses at limit	Tokens in responses at limit	At-limit responses rewarded
SmolLM3-3B	19.21	60.9%	0.8%	10.4%	1.9%
Qwen3.5-9B	16.60	51.5%	10.5%	63.6%	7.7%
Qwen3.5-35B-A3B	12.46	50.5%	6.4%	51.5%	5.2%

Where the training tokens go. Table 1 shows that about half of the response tokens in each GRPO run belong to groups whose eight rewards agree. These tokens are generated but carry zero advantage. On Qwen3.5-9B, the 10.5% of responses that reach the training limit account for 63.6% of all generated tokens, and only 7.7% of them earn a reward.

Evaluation. Sections 5 and 6 use 216 held-out problems from six 2025–2026 competitions (SMT, CMIMC, HMMT February and November, AIME, BRUMO). Section 4 uses a separate suite of 500 competition problems and reports the 216 as a secondary check. We sample 64 responses per problem at temperature 1 with a 4,096-token evaluation limit. If c_i of the 64 responses to problem i are correct, *coverage at 64* is $C_{64} = \frac{1}{n}\sum_i \mathbf{1}[c_i > 0]$ over the n problems, and pass@1 is $\frac{1}{n}\sum_i c_i/64$. For evaluation under a token budget $B = 16{,}384$, let $\ell_1, \ell_2, \ldots$ be the lengths of a problem's responses in sampled order. The responses *within the budget* are the first $K_B = \max\{k : \sum_{j \le k} \ell_j \le B\}$, and a response that would end past B does not count. Among them, a response is *finished* if it ends before the evaluation limit and contains an extractable final answer, and $F_B \subseteq \{1, \ldots, K_B\}$ denotes the finished responses. *Majority-vote accuracy*, S_B, scores the most frequent answer in F_B (ties go to the answer that appeared first), and *budgeted coverage*, C_B, asks whether any response in F_B is correct. Unfinished responses use tokens but cannot vote, and a problem without a finished response counts as wrong. Answer correctness uses an exact-match checker (Appendix B.3). *Supported coverage* further requires that an LLM judge, Muse Spark 1.3, label at least one correct response in F_B as supported. The judge sees only the problem and the response and applies a fixed rubric (Appendix H).

Uncertainty. We write $x \pm h$ for a paired difference x with a 95% bootstrap interval of half-width h over problems (10,000 resamples, with h the average of the two sides when the interval is slightly asymmetric). These intervals condition on the trained checkpoints and their sampled responses. When a problem's sampled responses run out before the budget and its outcome is not determined, we report the full range of possible values ("a to b") instead of an interval.

4 Harder problems raise the cost of unchanged problems

The responses to a hard training problem tend to be long. If training on it also makes the model's responses longer on other problems, it has an indirect cost as well. The model pays that cost on every other problem in the training dataset. The dataset-average response length combines both costs. To separate out the indirect cost, we measure response length on a fixed set of problems.

Design. We build two training sets of 3,072 problems each. Both sets contain the same 2,048 *shared problems* at the same positions in the training order. The other 1,024 positions are different in the two sets. In the *uniform mix*, we take problems sampled uniformly from the training pool. In the *hard mix*, we take problems on which the base model has a low success rate. Both mixes take the same number of problems from each source (see Appendix C.1). We train SmolLM3-3B on each set for 500 steps. One pass through a set takes 192 steps, so each run makes about 2.6 passes. Over the 500 steps, the shared problems appear 5,351 times in each run. Each time, the problem is at the same step and position in both runs. With eight responses each time, each mix has 42,808 responses to shared problems. Therefore, apart from sampling noise, a difference in response length on the shared problems shows how training on the 1,024 differing problems has changed the model.

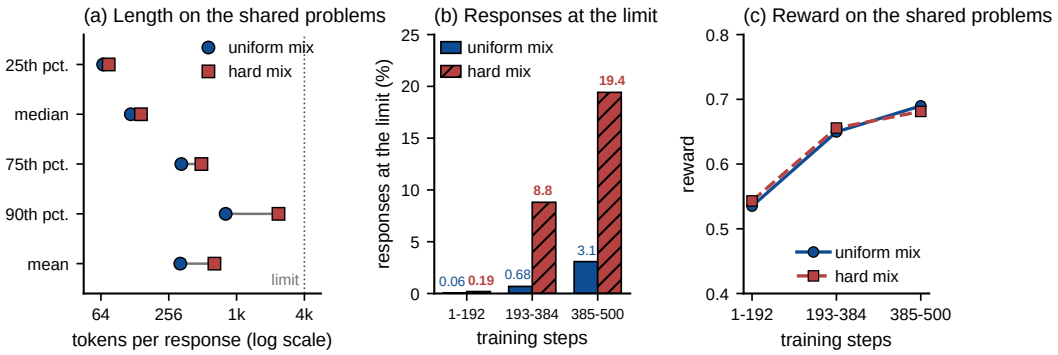


Figure 2: **The hard mix makes responses to the shared problems longer (SmolLM3-3B).** Each response under the hard mix is compared with the responses at the same step and position under the uniform mix. (a) Percentiles and mean of response length over steps 1–500, on a log scale. (b) Share of responses that reach the 4,096-token limit in each pass. (c) Mean training reward per group of eight responses in the first pass, the second pass, and the partial third pass.

The shared problems cost twice as many tokens. Over the 500 steps, responses to the shared problems average 650.9 tokens under the hard mix and 324.1 tokens under the uniform mix, an increase of 100.8%. The increase grows over training: it is 41.9% in the first pass, 109.9% in the second pass, and 124.9% in the partial third pass (Figure 1a).

The extra length moves from longer responses to responses that reach the limit. The hard mix lengthens long responses much more than typical ones (Figure 2a). The median length increases from 118 to 145 tokens, and the 90th percentile increases from 819 to 2,412 tokens. The source of the extra length changes over training (Figure 2b). In the first pass, fewer than 0.2% of the responses reach the 4,096-token limit under either mix, and the responses that end below the limit are 39.7% longer under the hard mix. In the second pass, 8.8% of the hard-mix responses reach the limit, against 0.7% under the uniform mix. In the third pass, the responses that end below the limit are only 5.9% longer under the hard mix, but 19.4% of its responses reach the limit, against 3.1% under the uniform mix. These responses account for 70.7% of the hard mix's tokens on the shared problems, although its median response is 201 tokens long. A shorter training limit acts directly on these responses (Section 5).

Training reward on the shared problems stays close. The mean reward on the shared problems is 61.9% under the hard mix and 61.5% under the uniform mix (Figure 2c). The paired difference is 0.33 ± 0.48 percentage points. In each pass, the two mean rewards differ by less than one point. The hard mix thus uses about twice as many tokens on the shared problems for a similar training reward.

The shared problems account for most of the added training cost. Over the full 500 steps of training, the hard mix generates 43.8 million response tokens whereas the uniform mix generates 19.2 million response tokens. Of the 24.6 million extra tokens generated by the hard mix, 14.0 million (56.9%) come from the shared problems and 10.6 million (43.1%) from the 1,024 differing problems. A differing problem's responses under the hard mix are 199.5% longer than the responses at the same position under the uniform mix, against 100.8% for a shared problem. But the shared problems fill two thirds of each training set, so their smaller increase accumulates more tokens in total. Hence, an estimate that counts only the responses to the hard problems misses 57% of the added cost.

The extra tokens do not give a clear gain on held-out problems. At step 500, the two models have a correct response among 64 attempts on almost the same number of the 500 held-out problems: 219 for the hard mix and 225 for the uniform mix. Under the 16,384-token budget, the hard mix has the higher majority-vote accuracy, 24.8% against 19.4% to 19.8%, but the uniform mix has the higher budgeted coverage, 32.6% against 26.6% (Appendix C). On the 216 problems of Section 5, majority-vote accuracy is 8.8% for the hard mix and 8.3% to 9.3% for the uniform mix, and the uniform mix again has the higher budgeted coverage, 15.3% to 16.2% against 11.1%. The order also

changes with the evaluation limit (Appendix C). The clear effect of the harder mix is therefore on training cost: it makes the unchanged problems more expensive to train on.

Implication. The cost of a training problem includes more than the length of its own responses. Training on hard problems changes the model, and the model then uses more tokens on the other problems, mostly in responses that reach the limit. Section 5 therefore tests the limit directly on Qwen3.5-9B: whether a shorter training limit lowers the cost without lowering majority-vote accuracy.

5 A SHORTER TRAINING LIMIT LOWERS COST AND IMPROVES BUDGETED ACCURACY

Some RLVR recipes allow long responses during training. DAPO, for example, generates up to 20,480 tokens per response and argues that longer responses give the model more room to explore (Yu et al., 2025). In our Qwen3.5-9B run with a 4,096-token training limit, the responses that reach this limit account for 63.6% of all generated tokens, yet only 7.7% of them earn a reward (Table 1). We therefore ask whether this run needs its 4,096-token limit. Surprisingly, halving the limit to 2,048 tokens lowers the training compute by 28% and raises majority-vote accuracy by 7.4 points. Under the longer limit, most of the model's evaluation answers do not finish within the evaluation limit.

Design. We train a second Qwen3.5-9B model with the recipe of Section 3 and a 2,048-token training limit. Everything else in the recipe is the same: the 3,072 problems and their order, the 192 updates, the optimizer, the reward, and the LoRA initialization. Both models are evaluated on the 216 problems with a 4,096-token evaluation limit and $B = 16{,}384$.

Training compute falls by 28%, and majority-vote accuracy rises by 7.4 points. The shorter limit halves the generated training tokens, from 16.60 to 8.54 million. The estimated training compute (Section 3, with $N = 9 \times 10^9$) falls by less, from 2.05 to 1.47 $\times 10^{18}$ FLOPs (28%), because the training pass also processes the prompts, 15.75 million tokens in both runs. Majority-vote accuracy rises from 36.1% to 43.5% (Table 2), a difference of 7.4 ± 4.9 points, while pass@1 rises by 7.3 ± 1.5 points and coverage at 64 by 12.5 ± 4.9 (Figure 3a). The gain comes from answers that finish. Because a truncated answer counts as wrong, pass@1 $= f\,a$, where f is the share of answers that finish within the evaluation limit and a is the accuracy of finished answers: $29.8\% \times 78.4\% = 23.4\%$ for the 4,096-token model and $47.8\% \times 64.1\% = 30.6\%$ for the 2,048-token model. The 2,048-token model's finished answers are less often correct, but far more of its answers finish. Under the budget, where only finished answers vote, it has a finished answer on 163 problems against 97. The pass@1 gain holds in each of the six contests (Appendix E).

Table 2: **The two Qwen3.5-9B runs.** Evaluation on 216 problems with a 4,096-token evaluation limit, and every vote under the budget is determined. Differences are 2,048 minus 4,096 in points unless stated, with 95% intervals where available.

	Training limit		
Measure	4,096	2,048	Difference
Training			
Estimated compute (10^{18} FLOPs)	2.05	1.47	-28%
Mean response length (tokens)	675.3	347.5	-48.5%
Reward, last 48 updates	0.796	0.783	-0.013
Evaluation, 64 attempts			
pass@1 (%)	23.4	30.7	$+7.3 \pm 1.5$
Coverage at 64, C_{64} (%)	53.2	65.7	$+12.5 \pm 4.9$
Answers reaching the evaluation limit (%)	70.2	52.2	-17.9 ± 2.0
Under the 16,384-token budget			
Majority-vote accuracy, S_B (%)	**36.1**	**43.5**	$\mathbf{+7.4 \pm 4.9}$
Problems without a finished answer	119	53	-66
Problems with a supported correct answer	74	94	$+20$

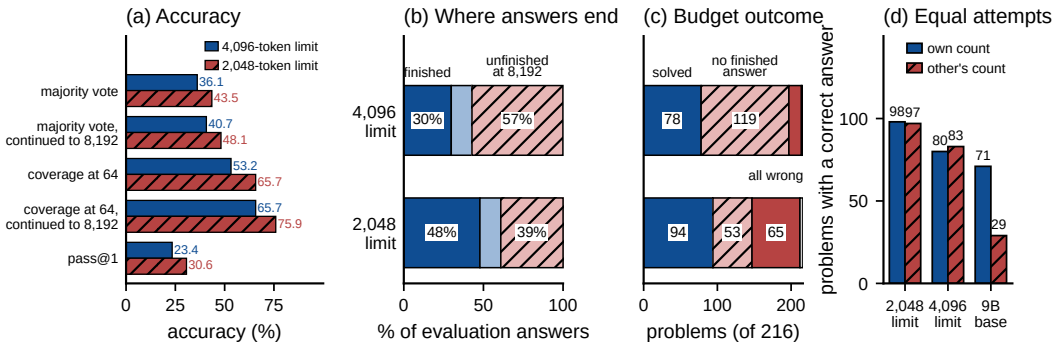


Figure 3: **The shorter training limit on Qwen3.5-9B.** (a) Majority-vote accuracy, coverage at 64, and pass@1 of the two models, at the 4,096-token evaluation limit and with truncated answers continued to 8,192 tokens. (b) How the 13,824 evaluation answers end. The light segments (13%) finish between 4,096 and 8,192 tokens when continued. (c) Outcome of the majority vote within the budget on the 216 problems. The thin segments at the right (2 and 4 problems) have a correct answer that is outvoted. (d) Problems with at least one correct answer when each model keeps its own number of attempts within the budget, or the other model's number. The base model is a control.

The gain survives continued answers, and the coverage advantage survives equal attempts. First, the evaluation limit truncates 70.2% of the 4,096-token model's answers, and a truncated answer counts as wrong. When every truncated answer of both models is continued from its saved prefix to 8,192 tokens, the majority-vote difference stays at 7.4 ± 5.1 points. Second, the 2,048-token model fits 9.00 attempts per problem into the budget and the 4,096-token model 7.05. At the 4,096-token model's number of attempts on every problem, the 2,048-token model still has a correct answer on 97 problems against 80, and at its own number, on 98 against 83. The same check lowers the 9B base model from 71 to 29 (Figure 3d).

The longer limit's extra length goes into answers that do not finish. When continued to 8,192 tokens, 57.1% of the 4,096-token model's 13,824 evaluation answers are still unfinished, against 39.1% (Figure 3b), and none of its 9,698 truncated answers is a repetition loop (Appendix E). Even with 8,192 tokens per answer, it covers 142 problems at 64 attempts, as many as the 2,048-token model covers with 4,096 tokens. Within the budget, it has no finished answer on 119 of the 216 problems, against 53, while only 17 of its problems have finished answers that are all wrong, against 65 (Figure 3c). The two models' finished answers come from different problems, so these counts do not compare answer quality. The 2,048-token model finishes on 66 more problems and solves 16 more.

Training signals do not reveal the problem. The 4,096-token run generates about twice the training tokens for a similar training reward, 0.796 against 0.783 over the last 48 updates. In those updates, 14.0% of its training responses reach the training limit, whereas 70.2% of its answers to the held-out problems do. Cutting the recorded responses at 2,048 tokens does not reproduce training under the shorter limit either: the cut removes 35.9% of the generated tokens, but also the reward of 605 of the 769 rewarded responses longer than 2,048 tokens, and it changes the direction of the update (Appendix D).

The length gap opens early and is carried by rewarded responses. Over the first 48 updates, training responses average 187.9 tokens under the 4,096-token limit and 147.6 under the 2,048-token limit. Over updates 49 to 96, they average 684.6 and 287.2. Rewarded responses grow as well: over the last 48 updates they average 362.5 and 211.6 tokens, and 5.8% of the rewarded responses of the 4,096-token run are longer than 2,048 tokens, so the shorter limit would have truncated them. This pattern agrees with the argument of Section 1: under the longer limit, a correct response that reaches its answer late gives a positive advantage to all of its tokens. We have not tested this mechanism directly.

The shorter answers are not weaker derivations at the problem level. The 2,048-token model has a supported correct answer on 94 problems and the 4,096-token model on 74, a difference of 9.3 ± 4.9 points, no smaller than the 8.3-point difference in budgeted coverage. Per answer, 8.5% of the 2,048-token model's correct answers are not supported, against 4.6%, a difference of 3.9 ± 2.5 points.

Scope. This result is one counterexample to the view that a longer training limit costs only compute. It comes from one model and two limits, and with more than 192 updates the 4,096-token model might learn to use its length.

6 Coverage and majority vote can give opposite rankings under the same token budget

With 16,384 tokens per problem, the SmolLM3-3B base model has at least one correct answer on 57 of the 216 problems, against 39 for its GRPO-trained version. Yet the majority answer over the same responses is correct on only 5 problems for the base model and on 18 for the trained model (Figure 4a). The two counts use the same budget and the same responses and differ only in what counts as solving a problem. Without a budget, at 64 attempts each, coverage ranks the models the other way: the trained model covers 78 problems against 43.

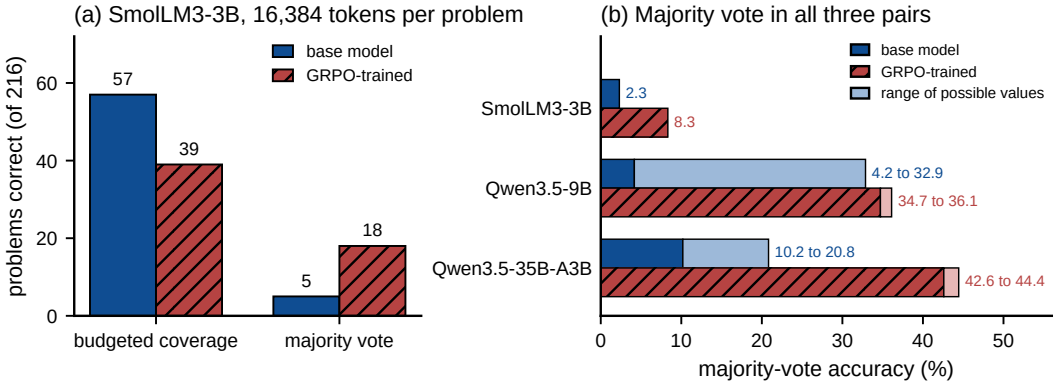


Figure 4: **Coverage and majority vote under the same budget.** (a) SmolLM3-3B base and GRPO-trained models at 16,384 tokens per problem: problems with a correct answer among the responses within the budget (budgeted coverage), and problems whose majority answer is correct. A response counts only if it ends within the budget, and only responses that finish within the 4,096-token evaluation limit with an extractable answer can vote. The base model has 256 saved responses per problem, so none runs out before the budget. (b) Majority-vote accuracy of each base model and its trained model. Where the 64 saved responses run out before the budget, the lighter segment spans the values over all possible outcomes and is not a confidence interval.

Majority vote favors the trained model in all three pairs. On SmolLM3-3B, the trained model's majority-vote accuracy is higher by 6.0 ± 3.7 points. On the Qwen models, the 64 saved responses of the base model run out before the budget on 62 problems (9B) and 23 problems (35B), so the differences are ranges, but both ranges lie on the trained side: $+1.85$ to $+31.94$ points on Qwen3.5-9B and $+21.76$ to $+34.26$ on Qwen3.5-35B-A3B (Figure 4b). These ranges describe the uncertainty from the missing responses. Budgeted coverage points the other way only on SmolLM3-3B: it favors the base there, is unresolved on Qwen3.5-9B (trained minus base, -8.80 to $+4.17$ points), and favors the trained model on Qwen3.5-35B-A3B ($+2.31$ to $+4.63$).

Budgeted coverage depends on how many attempts fit. SmolLM3-3B base responses average 153 tokens and trained responses 1,871, so a median of 113 base attempts fits the budget against 9 trained attempts. When each base model keeps, on every problem, only as many saved responses as its trained model fits into the budget, its coverage falls from 71 to 29 problems on Qwen3.5-9B, from 91 to 49 on Qwen3.5-35B-A3B, and from 43 to 15 on SmolLM3-3B with 64 saved responses

(Table 11). The same check moves the two models of Section 5 by at most three problems. This comparison describes the saved responses and does not isolate a causal effect of response length. It agrees with the observation that base models can overtake RL-trained models at large attempt counts (Yue et al., 2025).

Requiring a supported derivation also favors the trained model. Under supported coverage (Section 3), the SmolLM3-3B base covers 3 problems and the trained model 16, a difference of 6.0 ± 3.0 points. On Qwen3.5-35B-A3B the base covers at most 61 problems against 94, and on Qwen3.5-9B the order is unresolved (37 to 82 against 74). CoT-Pass@k applies the same requirement at a fixed number of attempts (Wen et al., 2025), and here it is applied under a token budget. The audit uses one judge without human labels, and the judge sees response length and style.

Extra problems covered at 64 attempts can fall outside the budget. On Qwen3.5-35B-A3B, a model trained with a random subset of rewards removed (R-M in Appendix F) covers 144 problems at 64 attempts, against 137 for the full-reward model, a difference of 3.2 ± 4.2 points. None of the 14 problems that only it covers has a correct response within the budget, and its majority vote is correct on 3 to 12 fewer problems. With all 64 responses and no budget, the two majority votes tie at 119 problems, so the reversal comes from the budget rather than from the switch to voting. Choosing between these models by coverage at 64 attempts would select the one that returns fewer correct answers under the budget (Appendix G).

7 Discussion and conclusion

What the extra tokens bought. In all three experiments, more tokens did not mean more correct answers. The harder mix doubled the length of responses to the unchanged problems without a clear gain on held-out problems. The 4,096-token limit spent its extra length on answers that do not finish, and halving it lowered training compute by 28% and raised majority-vote accuracy by 7.4 points. In both training experiments, the training reward of the two conditions stayed close and did not reveal these differences. At evaluation, the SmolLM3-3B base model's many short attempts found a correct answer on more problems than its trained version (57 against 39) but returned a correct majority answer on fewer (5 against 18).

What to measure. When a data choice changes, measure response length on the problems that stay the same, because an average over the training set also changes with the mix itself. Report training compute and not only generated tokens: halving the generated tokens saved 28% of the compute. Measure truncation on held-out problems at the evaluation limit, since truncation during training can understate it (14.0% against 70.2% for the 4,096-token run). Compare models under a common token budget and a stated per-response limit, and report the answer each model returns together with whether any attempt is correct, since the two can rank models in opposite orders.

Limitations. Each main result rests on one model: the shared-problem result on SmolLM3-3B, and the limit result on Qwen3.5-9B with two limits and 192 updates, which do not locate a best limit. We did not compare the shorter limit with other length controls, such as overlong reward shaping (Yu et al., 2025) or length penalties. The derivation audit uses one LLM judge without human labels, and we did not check the evaluation problems against the models' pretraining data.

Conclusion. In our experiments, the number of rollout tokens was a poor guide to what RLVR training produced. We therefore suggest judging a training choice by what it costs on the problems that stay the same and by the answers the trained model returns under a fixed inference budget.

REPRODUCIBILITY STATEMENT

Appendix B gives the training recipes, data partitions, reward and evaluation rules, the token-budget rule, and the uncertainty calculations. Appendices C–H give the details of every experiment in the main text, including the reward-deletion study and the derivation-audit rubric and calibration. The supplementary material contains the aggregate results behind every table and figure and the Python scripts that draw the figures.

AI USE STATEMENT

Language models assisted with code development and review, literature search, analysis, and writing. The derivation audit in Section 6 uses a language model as the judge. The authors designed the experiments and checked all reported numbers against the experiment records.

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.04697.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning, 2025. URL https://arxiv.org/abs/2504.03380. EACL 2026 main conference (arXiv comment).

Longtian Bao, Jianyou Wang, Yang Zhang, Youze Zheng, and Ramamohan Paturi. Question begets question: Self-evolving curriculum for reinforcement fine-tuning on competition mathematics, 2026. URL https://arxiv.org/abs/2608.01522.

Abdelaziz Bounhar, Hadi Abdine, Evan Dufraisse, Ahmad Chamma, Amr Mohamed, Dani Bouch, Michalis Vazirgiannis, and Guokan Shang. Shorter but not worse: Frugal reasoning via easy samples as length regularizers in math rlvr, 2025. URL https://arxiv.org/abs/2511.01937.

David Brandfonbrener, Simon Henniger, Sibi Raja, Tarun Prasad, Chloe Loughridge, Federico Cassano, Sabrina Ruixin Hu, Jianang Yang, William E. Byrd, Robert Zinkov, and Nada Amin. Vermcts: Synthesizing multi-step programs using a verifier, a large language model, and tree search, 2024. URL https://arxiv.org/abs/2402.08147.

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, 2024. URL https://arxiv.org/abs/2407.21787.

Weize Chen, Jiarui Yuan, Tailin Jin, Ning Ding, Huimin Chen, Zhiyuan Liu, and Maosong Sun. The overthinker's diet: Cutting token calories with difficulty-aware training, 2025. URL https://arxiv.org/abs/2505.19217.

Yue Cheng, Jiajun Zhang, Xiaohui Gao, Weiwei Xing, Zheng Wang, and Zhanxing Zhu. Mechanistically interpreting the role of sample difficulty in RLVR for LLMs, 2026. URL https://arxiv.org/abs/2605.28388.

Yuyang Ding, Chi Zhang, Juntao Li, Haibin Lin, Xin Liu, and Min Zhang. FAPO: Flawed-aware policy optimization for efficient and reliable reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jhqqoimoWt.

Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. Concise reasoning via reinforcement learning, 2025. URL https://arxiv.org/abs/2504.05185.

Thomas Foster, Anya Sims, Johannes Forkel, and Jakob Foerster. Lilo: Learning to reason at the frontier of learnability. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 48941–48974. Curran Associates, Inc., 2025. doi: 10.52202/

085713-1634. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/45d5823950cab9a88e10c6e8dbff351c-Paper-Conference.pdf.

Rodrigo Guedes de Souza and Alison R. Panisson. Who thinks best depends on how long you let them: Budget-dependent rankings in LLM evaluation, 2026. URL https://arxiv.org/abs/2608.12150.

Daya Guo, Dejian Yang, Haowei Zhang, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

Andre He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening, 2025. URL https://arxiv.org/abs/2506.02355.

HMMT. HMMT (Harvard-MIT mathematics tournament), 2026. URL https://www.hmmt.org. Accessed 2026-09-25.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of LLMs via reinforcement learning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=V51gPu1uQD.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2024. URL https://arxiv.org/abs/2411.15124.

Ziniu Li, Congliang Chen, Tianyun Yang, Tian Ding, Ruoyu Sun, Ge Zhang, Wenhao Huang, and Zhi-Quan Luo. Knapsack RL: Unlocking exploration of LLMs via optimizing budget allocation, 2025. URL https://arxiv.org/abs/2509.25849.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective, 2025. URL https://arxiv.org/abs/2503.20783.

Mathematical Association of America. MAA invitational competitions (AIME), 2026. URL https://maa.org/maa-invitational-competitions/. Accessed 2026-09-25 via the Internet Archive snapshot of 2026-09-14.

Michael Noukhovitch, Hamish Ivison, Nathan Lambert, and Aaron Courville. Learning to solve hard problems in RL for LLMs by never giving up, 2026. URL https://arxiv.org/abs/2609.13443.

Benjamin Pikus, Pratyush Ranjan Tiwari, and Burton Ye. Hard examples are all you need: Maximizing GRPO post-training under annotation budgets, 2025. URL https://arxiv.org/abs/2508.14094.

Amrith Setlur, Matthew Y. R. Yang, Charlie Snell, Jeremy Greer, Ian Wu, Virginia Smith, Max Simchowitz, and Aviral Kumar. e3: Learning to explore enables extrapolation of test-time compute for LLMs, 2025. URL https://arxiv.org/abs/2506.09026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Akifumi Wachi, Hirota Kinoshita, Shokichi Takakura, Rei Higuchi, and Taiji Suzuki. A relative-budget theory for reinforcement learning with verifiable rewards in large language model reasoning, 2026. URL https://arxiv.org/abs/2602.01523.

Junlin Wang, Siddhartha Jain, Dejiao Zhang, Baishakhi Ray, Varun Kumar, and Ben Athiwaratkun. Reasoning in token economies: Budget-aware evaluation of LLM reasoning strategies, 2024. URL https://arxiv.org/abs/2406.06461.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs, 2025. URL https://arxiv.org/abs/2506.14245.

Chen Yang, Xianyang Zhang, and Jun Chen. Rlvr is a kernel, not a function: Statistical inference for pass@k crossovers, 2026. URL https://arxiv.org/abs/2609.22547.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model?, 2025. URL https://arxiv.org/abs/2504.13837.

Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for LLM reasoning via selective rollouts, 2025. URL https://arxiv.org/abs/2506.02177.

Todd Zhou. When rlvr shrinks the reasoning boundary: Diagnosing pass@k inversion, 2026. URL https://arxiv.org/abs/2607.20543.

Yuzhen Zhou, Jiajun Li, Yusheng Su, Gowtham Ramesh, Zilin Zhu, Xiang Long, Chenyang Zhao, Jin Pan, Xiaodong Yu, Ze Wang, Kangrui Du, Jialian Wu, Ximeng Sun, Jiang Liu, Qiaolin Yu, Hao Chen, Zicheng Liu, and Emad Barsoum. APRIL: Active partial rollouts in reinforcement learning to tame long-tail generation, 2025. URL https://arxiv.org/abs/2509.18521.

A CLAIMS AND EVIDENCE

Table 3 lists each claim of the main text with the evidence behind it and the conditions under which it holds. The remaining appendices give the details. Unless stated otherwise, accuracies are percentages and differences are percentage points. As in the main text, $x \pm h$ gives a paired difference with the half-width of its 95% bootstrap interval over problems, and "a to b" gives the range of values over all possible outcomes where the saved responses run out before the budget. Such ranges are not confidence intervals. All intervals condition on the trained checkpoints and their sampled responses.

The appendices also report experiments that do not enter the main claims: the reward-deletion study (Appendix F) and budgeted results of all deletion checkpoints (Appendix G).

B TRAINING AND EVALUATION PROTOCOLS

B.1 TRAINING RECIPES AND DIFFICULTY LABELS

For the Qwen runs, LoRA is applied to attention, MLP, and unembedding parameters with initialization seed 0. The stop string is `\nProblem:`, and checkpoints are saved every 32 updates. All Qwen runs share the same recipe.

The SmolLM3 deletion study uses 8,000 prompts, of which 3,789 are assigned to the hard cohort and 2,998 to the medium cohort from the reference run's success rates. The shared-problem training sets define difficulty from the base model's success rate before training. The Qwen deletion cohorts are derived from each model's 16-response probe of the 3,072-prompt pool. Difficulty is therefore model-specific and is not pooled across models.

B.2 PROBLEM SUITES AND SCORING

The 216-problem evaluation suite combines SMT 2025 (53), CMIMC 2025 (40), HMMT February 2026 (33), AIME 2026 (30), HMMT November 2025 (30), and BRUMO 2026 (30). Thus 123 problems are from 2025 and 93 from 2026. We did not check these problems against the pretraining data of any of the models. Appendix E splits the limit result by contest year.

The 500-problem prompt-mix suite is separate: HMMT 96, AIME 79, Purple Comet 62, BMT 60, PUMaC 52, SMT 49, CMIMC 31, and other competitions 71, spanning earlier years through 2025. In the 216-problem suite, 134 reference answers are integers, including 105 from 0 to 999; 34 are fractions and 48 are other expressions.

The evaluation samples 64 responses per problem with a 4,096-token response limit, temperature 1, top-p 1, and the training few-shot prefix. Answer correctness uses an exact-match checker with fixed normalization and extraction rules; it does not check derivations.

B.3 BUDGET BOUNDARY AND MAJORITY VOTING

Let $L_{q,j}$ be the generated length of response j to problem q, in its saved order. The responses within budget B are

$$\mathcal{J}_q(B) = \left\{ j : \sum_{k \le j} L_{q,k} \le B \right\}. \tag{2}$$

A response cannot vote if it reaches C_{eval} or lacks an extractable normalized answer, but its length still counts toward the sum. Among the responses that can vote, the most frequent answer is returned, and ties go to the answer that appeared first. A problem with no such response counts as wrong. Budgeted coverage checks whether any of these responses has a correct answer, without requiring the vote to choose it.

Under Equation 2, a response that starts inside the budget but would end past it does not count, so the responses within the budget use at most B tokens on every problem.

If the saved responses of a problem run out while another response could still fit, its outcome is open unless the saved responses already determine it. We then report the range of values over all possible

Table 3: Main claims, their evidence, and their scope.

Claim	Evidence	Scope and conditions	Details
Harder problems raise the cost of unchanged problems	Mean length on 2,048 shared problems $+100.8\%$ (324 $\rightarrow$ 651 tokens); median, 90th percentile and below-limit mean also rise; reward 0.33 ± 0.48 points	SmolLM3-3B, full-parameter verl, 500 updates	C
The added length starts as longer responses below the limit and grows into responses at the limit	First pass: responses below the limit $+39.7\%$, fewer than 0.2% at the limit; third pass: below the limit $+5.9\%$, at the limit 19.4% against 3.1%	Same run; measured in each pass through the training set	C
The shared problems account for most of the added training cost	14.0 of 24.6 million added tokens on shared problems	Same run	C
The harder mix has no clear held-out gain	Majority vote favors the hard mix, budgeted coverage the uniform mix, and the order depends on the evaluation limit	500 held-out problems, step 500	C
Cutting stored responses does not predict training under a shorter limit	Cut at 2,048: -35.9% tokens, -4.3% total advantage, retained-token cosine 0.65 vs. 0.95	Post hoc, coefficient level, not gradients	D
A shorter training limit lowers cost and raises budgeted accuracy	Training compute -28%; majority vote $+7.4 \pm 4.9$; pass@1 $+7.3 \pm 1.5$; C_{64} $+12.5 \pm 4.9$	Qwen3.5-9B, LoRA, 192 updates	E
The gain holds under a longer evaluation limit	C_{64} gap 10.2 ± 4.4 at 8,192; vote $+7.4 \pm 5.1$	Same checkpoints, all truncated responses continued	E
The coverage advantage persists at equal attempt counts	At equal attempt counts, 97 against 80 and 98 against 83 problems with a correct answer; the same check lowers the base model from 71 to 29	Assumes the order of sampled answers carries no information	E
The longer limit's extra length goes into answers that do not finish	57.1% against 39.1% of evaluation answers unfinished at 8,192 tokens; no finished answer within the budget on 119 against 53 problems	Original pair, 13,824 answers per model	E
The shorter answers are not weaker derivations at the problem level	Supported problems 94 against 74 ($+9.3 \pm 4.9$ points); correct answers not supported 8.5% against 4.6%	One LLM judge, no human labels	H
The shorter-limit model finishes more often	47.8% vs. 29.8% of responses finish; accuracy of finished responses 64% vs. 78%; truncated responses are rarely loops	64 responses per problem, 4,096-token evaluation limit	E
Coverage and majority vote rank a base model and its trained model in opposite orders	SmolLM3-3B at 16,384 tokens: C_B base 57, GRPO 39; majority vote base 5, GRPO 18	256 base responses per problem; 113 vs. 9 median attempts	G
Majority vote favors the trained model in all three pairs	SmolLM3-3B $+6.0 \pm 3.7$; Qwen3.5-9B $+1.85$ to $+31.94$; Qwen3.5-35B-A3B $+21.76$ to $+34.26$	Qwen ranges where base responses run out	G
Requiring a supported derivation also favors the trained model	Supported coverage: base 3, GRPO 16 ($+6.0 \pm 3.0$); 35B-A3B also favors GRPO; 9B unresolved	One LLM judge, no human labels	H
The evaluation limit can reverse a data comparison	Hard vs. uniform C_{64}: 43.8 vs. 45.0 at 4,096; 59.2 vs. 48.8 at 8,192	500 problems, step 500	C

Table 4: Training recipes. These differences preclude interpreting the experiments as a controlled comparison of model size.

Setting	SmolLM3-3B-Base	Qwen3.5-9B / 35B-A3B-Base
Update method	Full-parameter training, verl	Rank-32 LoRA
Updates	500	192
Problems per update	16	16
Responses per problem	8	8
Training response limit	4,096	4,096; 2,048 in limit tests
Advantage	Reward minus group mean	Reward minus group mean
Loss aggregation	Token mean	Importance sampling, coefficients scaled by $1/T_s$
KL penalty / advantage std.	Neither	Neither
Optimizer	Learning rate 10^{-6}	AdamW, 10^{-5}; $\beta = (0.9, 0.95)$, no weight decay or clipping
Sampling	Temperature 1; rollout seed 42	Temperature 1, top-p 1; no fixed seed
Training set	8,000 positions in deletion study; 3,072 repeated in mix study	3,072 positions, each used once

outcomes of these problems, computed problem by problem and paired across conditions, rather than assigning them zero or a midpoint. A paired bootstrap interval over all 216 problems is reported only when every outcome is determined in both conditions.

For completeness, C_{64} is the fraction of problems with at least one correct response among 64. Pass@1 is the mean per-problem correct-response fraction. For n sampled responses and c correct, the usual estimator of pass@k is $1 - \binom{n-c}{k} / \binom{n}{k}$. Majority voting over all 64 saved responses, without the token stopping rule, is denoted S_{64}. It is different from both C_{64} and S_B.

B.4 Uncertainty

The bootstrap resamples the same problem indices jointly across the compared conditions 10,000 times (seed 20260917). The shared-problem reward analysis instead resamples the 2,048 shared problems as clusters, keeping all appearances of a problem together (seed 20260925). Intervals after the derivation audit treat the judge labels as fixed. The main text reports $x \pm h$ with h the average of the two sides of the percentile interval; the two sides of every reported interval differ by at most about one point.

The majority-vote readout of the training-limit comparison uses 54 additional responses drawn after the first sampling round (Appendix E).

C Shared-problem experiment

C.1 What the two training sets hold fixed

Both training sets contain 3,072 positions. The 2,048 shared problems occupy the same positions in both. The other 1,024 positions take 301, 702, and 21 problems from the three sources, drawn either uniformly from the pool or from the hard problems. A third run, the medium mix, fills them with medium-difficulty problems. One pass through a training set takes 192 steps, and the training order then repeats. Over 500 steps, the shared problems appear 5,351 times, each time with eight responses, which gives 42,808 responses per mix. Training uses rollout seed 42 in every run. Checks on the training logs confirm that each appearance of a shared problem has the same problem at the same step and position in both runs.

C.2 Length distribution in each pass

Table 5 shows how the extra length changes from pass to pass. The mean length of the responses below the limit is 39.7%, 29.5%, and 5.9% higher under the hard mix in the three passes. The share of responses at the limit is 0.19%, 8.81%, and 19.42% under the hard mix, against 0.06%, 0.68%,

Table 5: Response length on the shared problems (response-mask positions). "At limit" is the share of responses of exactly 4,096 tokens; "below limit" covers all other responses.

		All responses				At limit	Below limit	
Steps	Mix	Mean	Median	75th pct.	90th pct.	(%)	Mean	Median
1–500	Uniform	324.1	118	331	819	1.01	285.8	117
	Hard	650.9	145	499	2,412	8.00	351.4	128
1–192	Uniform	180.6	84	167	437	0.06	178.2	84
	Hard	256.2	95	243	694	0.19	248.9	95
193–384	Uniform	359.7	150	466	894	0.68	334.0	148
	Hard	755.1	183	716	3,159	8.81	432.5	159
385–500	Uniform	500.2	158	547	1,220	3.08	386.0	151
	Hard	1,125.0	201	1,594	4,096	19.42	408.8	150

and 3.08% under the uniform mix. Responses at the limit account for 3.0%, 47.8%, and 70.7% of the hard mix's tokens on the shared problems, against 1.4%, 7.8%, and 25.2% of the uniform mix's.

In total, the hard mix generates 43.83 million response-mask tokens over 500 steps and the uniform mix 19.21 million. Of the 24.62 million additional tokens, 13.99 million fall on the shared problems and 10.63 million on the 1,024 differing problems. Per response, the differing problems cost 199.5% more under the hard mix and the shared problems 100.8% more. The per-pass increases on the shared problems are 41.9%, 109.9%, and 124.9%.

C.3 Training reward on the shared problems

The mean reward per group on the shared problems is 0.6185 under the hard mix and 0.6152 under the uniform mix. The paired difference, with problems resampled as clusters, is 0.33 ± 0.48 points. In the first, second, and partial third pass, the rewards under the hard and uniform mixes are 0.5430 and 0.5350, 0.6556 and 0.6496, and 0.6813 and 0.6899.

C.4 Held-out results

Table 6: Held-out results at step 500 on 500 competition problems, 4,096-token evaluation limit, $B = 16{,}384$, 192 responses per problem. Majority-vote ranges come from problems whose saved responses run out before the budget. "Finished" counts problems with at least one finished response within the budget.

Mix	C_{64} (problems)	Majority vote (%)	Budgeted coverage (%)	Finished	Undetermined votes
Uniform	225	19.4 to 19.8	32.6	492	2
Hard	219	24.8	26.6	263	0
Medium	244	20.0 to 21.0	38.8	—	5

The hard mix has the higher majority-vote accuracy (the difference lies between $+5.0$ and $+5.4$ points) and the lower budgeted coverage (-6.0 points). Its checkpoint answers the held-out problems at much greater length, 3,380 against 1,167 tokens on average, and 67% of its responses reach the evaluation limit, so it has a finished response within the budget on only 263 of the 500 problems. On the 495 problems determined in all three mixes, the mean number of qualifying candidates within the budget is 2.9 for the hard mix, 20.2 for the uniform mix, and 24.8 for the medium mix. Writing majority-vote accuracy as $S = R + V$, with R the accuracy of a uniformly chosen candidate and V the gain from voting, the hard-minus-uniform difference of 5.05 ± 3.33 points splits into 7.34 ± 2.55 points in R and -2.28 ± 1.77 points in V: the hard mix has few candidates, but they are more often correct, and voting helps it less. The medium mix is not distinguished from the uniform mix (1.21 ± 2.53 points on the same 495 problems).

On the 216 problems of Section 5, evaluated with 64 responses per problem under the same limit and budget, majority-vote accuracy is 8.80% for the hard mix and 8.33% to 9.26% for the uniform mix, whose saved responses leave two votes undetermined. Budgeted coverage is 11.11% against 15.28%

to 16.20%. On the 214 problems determined for both mixes, the hard-minus-uniform difference is -5.14 ± 4.67 points in budgeted coverage and -0.47 ± 3.74 points in majority-vote accuracy.

The evaluation limit changes the hard-versus-uniform coverage comparison. At step 192, the difference is 1.60 ± 3.20 points at a 4,096-token limit and 1.80 ± 3.20 at 8,192. At step 500, it is -1.20 ± 3.70 and $+10.40 \pm 4.00$ points. The change between the two limits grows from step 192 to step 500 by 11.40 ± 3.60 points; counting any response with an extractable answer instead of only finished responses gives 8.20 ± 3.20 points. The interaction is not monotone over intermediate checkpoints: it is $+7.8$, $+12.2$, and $+10.4$ points at steps 320, 448, and 500.

D Token accounting and retrospective truncation

D.1 Where the coefficients are nonzero

In the implemented loss, each token of response i in update s carries the coefficient

$$c_{i,t} = A_i / T_s, \qquad T_s = \sum_{j \text{ in } s} L_j, \tag{3}$$

before the policy-ratio and parameter-derivative factors; $c_{i,t}$ is not a gradient. In a group with mixed rewards, incorrect responses carry negative coefficients. When all rewards of a group agree, its coefficients are zero, but its responses have already been generated and still enter the denominator T_s.

The Qwen runs with the 4,096-token limit generate 16.60 million response tokens (9B) and 12.46 million (35B-A3B). Of these, 51.5% and 50.5% belong to zero-advantage groups, and 63.6% and 51.5% belong to responses at the training limit; the two categories overlap. Responses at the limit are 10.5% and 6.4% of all responses, and 7.7% and 5.2% of them are rewarded. The SmolLM3-3B run uses 19.21 million response-mask positions, 60.9% of them in zero-advantage groups; 0.8% of its responses reach the limit and account for 10.4% of the tokens. In this run, 2,776 of 8,000 prompt groups have mixed rewards. The 2,000 competition-source groups carry about 13% of the absolute advantage, and 79% of them have no correct response.

D.2 Cutting stored responses at 2,048 tokens

We cut every recorded Qwen training response at 2,048 tokens, grade the retained prefix with the training grader, and recompute all group advantages. The cut removes 35.9% of the generated tokens for 9B and 29.9% for 35B-A3B. The total absolute advantage, $M = \sum_i |A_i|$, falls by only 4.3% and 2.6%. The sum of the absolute changes of individual advantages, $\sum_i |A'_i - A_i|/M$, is 27.8% and 20.3%. For comparison, re-estimating only each group's average from a bootstrap resample of its eight recorded rewards changes this sum by 32.1% and 32.4%, so the size of the change is comparable to baseline noise. Its direction is not (Figure 5b). On the retained token positions, the mean per-step cosine between the original and the recomputed coefficient vectors is 0.649 for 9B and 0.746 for 35B-A3B, against 0.949 for both under baseline resampling and 1 when the rewards are kept. The truncation moves credit between responses in a length-correlated way: a correct response that loses its answer at the cut lowers its group's average and raises the advantages of the other responses.

Among rewarded responses longer than 2,048 tokens, 605 of 769 (9B) and 443 of 508 (35B-A3B) lose their reward on the retained prefix, while 13 and 2 unrewarded responses gain one. The audit regrades 3,346 and 2,189 long responses; an independent reimplementation reproduces 144 reported values without a mismatch and all prefix grades. The cosine comparison excludes updates in which a coefficient vector is zero, leaving 190 updates for each model.

Cutting the stored 9B responses at 2,048 tokens would remove 35.9% of the generated tokens. The 2,048-token run generates 48.5% fewer tokens, 8.54 against 16.60 million, because the model trained under the shorter limit also writes shorter responses below the limit. The two numbers differ because training under a shorter limit changes the policy, the rewards, and the response lengths together, which is why Section 5 trains a new model instead of relying on this audit.

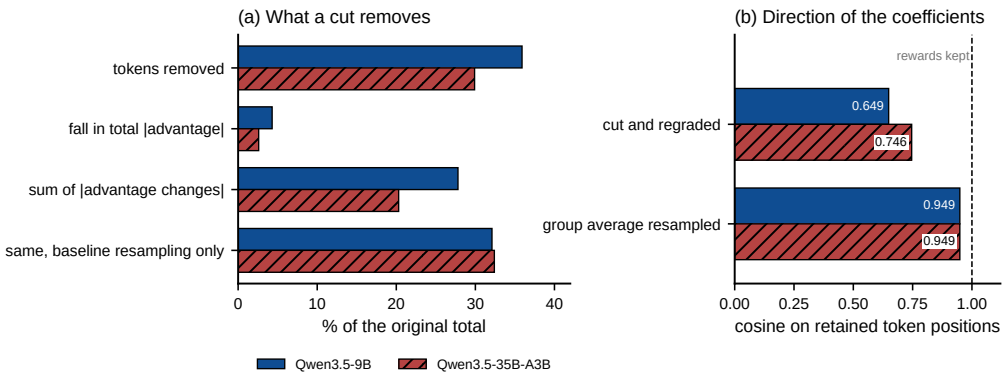


Figure 5: **Retrospective truncation of the recorded Qwen training responses at 2,048 tokens.** (a) Each quantity as a share of its original total; the last row resamples only the group average and keeps all responses. (b) Mean per-step cosine between the original and the recomputed coefficient vectors on the same retained token positions, over 190 comparable updates. These are coefficient comparisons, not gradient comparisons.

E SHORTER TRAINING LIMIT: DETAILS AND CHECKS

E.1 SAMPLING

Table 7: Majority-vote readouts of the original comparison as sampling proceeds. Ranges come from problems whose saved responses run out before the budget.

Sampling	4,096 limit (%)	2,048 limit (%)	Difference (points)
Planned (64 responses and one top-up round)	35.19 to 36.11	43.06 to 43.52	$+6.94$ to $+8.33$
With 54 additional responses	36.11	43.52	7.41 ± 4.86

With 64 responses per problem, the votes of three problems of the 4,096-token model and five of the 2,048-token model are not determined, and one more sampling round reduces them to two and one. The saved responses of the remaining three problems end 377, 714, and 1,201 tokens short of the budget. We drew 14, 9, and 31 further responses for them, 54 in total, which determines every vote. The readout before them already places the difference between $+6.9$ and $+8.3$ points; the additional responses determine every vote and make a paired interval possible.

Table 8: Training tokens and estimated training compute of the original comparison. Compute counts $2N$ FLOPs per uncached input or generated token and $6N$ per training-pass token, with $N = 9 \times 10^9$ parameters; cached input tokens are not recomputed. Attention FLOPs, which grow with length in 8 of the 32 layers, are ignored, so the estimate understates the saving. Counting $4N$ per training-pass token, as for LoRA without gradients for the frozen weights, gives a saving of 29.5% instead of 28.2%. Training-pass tokens are 24,576 fewer than the input and generated tokens together, one per training response. Evaluation is not included.

Quantity	4,096 limit	2,048 limit
Uncached input tokens	496,330	496,330
Cached input tokens	15,280,006	15,280,006
Generated response tokens	16,595,555	8,539,778
Training-pass tokens	32,347,315	24,291,538
Estimated training compute (10^{18} FLOPs)	2.05	1.47
Mean training response (tokens)	675.3	347.5
Mean evaluation response (tokens)	3,332.7	2,857.5

On the 64 responses per problem, the shorter limit raises coverage at 64 attempts by 12.50 ± 4.86 points and pass@1 by 7.26 ± 1.54 points; budgeted coverage rises from 37.0% to 45.4%. Splitting

the completed votes by contest year gives the same direction on the 2025 and the 2026 contests (Table 9), and pass@1 and coverage at 64 are higher in each of the six contests (Table 10).

Table 9: Majority votes of the original comparison split by contest year. Intervals resample problems within each year.

Contest year	Problems	4,096 limit correct	2,048 limit correct	Difference (points)
2025	123	41	50	7.32 ± 7.32
2026	93	37	44	7.53 ± 5.92

Table 10: Differences between the 2,048- and the 4,096-token training limit by contest, original runs, in points with 95% intervals over problems.

Contest	Problems	pass@1	Coverage at 64
AIME 2026	30	+9.17 [+4.90, +13.80]	+3.33 [−6.67, +13.33]
BRUMO 2026	30	+2.71 [−0.16, +5.83]	+6.67 [+0.00, +16.67]
HMMT February 2026	33	+3.98 [+1.80, +6.82]	+21.21 [+9.09, +36.36]
CMIMC 2025	40	+7.54 [+4.26, +11.48]	+12.50 [+0.00, +25.00]
HMMT November 2025	30	+9.38 [+5.16, +14.27]	+16.67 [+3.33, +30.00]
SMT 2025	53	+9.40 [+6.16, +12.88]	+13.21 [+5.66, +22.64]
2025 contests	123	+8.79 [+6.68, +11.05]	+13.82 [+7.32, +20.33]
2026 contests	93	+5.24 [+3.28, +7.34]	+10.75 [+4.30, +18.28]

E.2 Equal numbers of attempts

Under the budget, the 2,048-token model fits 9.00 attempts per problem and the 4,096-token model 7.05. To remove this difference, we keep, on each problem, as many of a model's saved answers as the other model fits into the budget, and count the problems with at least one correct answer (Table 11). The check assumes that the order of a problem's sampled answers carries no information, which holds because the answers are sampled independently. Changing the number of attempts moves the two limits by zero to three problems, while it moves each base model by 28 to 42 problems.

Table 11: Problems (of 216) with at least one correct answer when a model keeps, on each problem, as many of its saved answers as the first or the second model of the pair fits into the 16,384-token budget. Bold: each model's own count.

Answers of	At the first model's counts	At the second model's counts
4,096-token limit	**80**	83
2,048-token limit	97	**98**
Qwen3.5-9B base	**71**	29
Qwen3.5-9B, 4,096-token limit	109	**80**
Qwen3.5-35B-A3B base	**91**	49
Qwen3.5-35B-A3B, trained	127	**101**
SmolLM3-3B base	**43**	15
SmolLM3-3B, trained	78	**39**

E.3 Finishing and truncated responses

Of the 13,824 evaluation responses per model (64 for each of 216 problems), 4,126 finish within the 4,096-token evaluation limit for the 4,096-token model and 6,605 for the 2,048-token model. Of the finished responses, 3,233 (78.4%) and 4,237 (64.1%) are correct. The 2,048-token model covers 29 problems that the 4,096-token model does not, and the 4,096-token model covers 2 that the 2,048-token model does not, and its per-problem pass@1 is higher on 123 problems and lower on 13.

To check what the truncated responses contain, we ran two repetition detectors on every truncated evaluation response (Table 12). The strict detector flags a response whose last 800 characters match,

to at least 90%, the same text shifted by at most 1,200 characters. The weaker detector flags any 80-character window that occurs twice. Truncated responses of the trained models are rarely loops; the base model's truncated responses loop more often.

Table 12: Truncated evaluation responses of the Qwen3.5-9B models at the 4,096-token limit. "Extractable answer" counts truncated responses that contain an answer the extractor can read; the strict evaluation rule still scores them incorrect.

Model	Truncated / 13,824	Strict loop	Repeated 80-character window	Extractable answer
Base	457	9 (2.0%)	15 (3.3%)	253 (55.4%)
4,096 limit	9,698	0	51 (0.5%)	900 (9.3%)
2,048 limit	7,219	1	21 (0.3%)	634 (8.8%)

E.4 All truncated evaluation responses continued to 8,192 tokens

Every evaluation response of the two original checkpoints that reaches the 4,096-token limit is continued from its exact saved prefix to at most 8,192 tokens: 9,698 responses of the 4,096-token model and 7,219 of the 2,048-token model. The 64 responses per problem are kept, so coverage at 64 attempts is measured on the same attempts at both limits.

Table 13: Continuation to 8,192 tokens. Majority-vote accuracy uses the 16,384-token budget with the continued responses; every vote is determined.

Quantity	4,096 limit	2,048 limit
Problems covered at 64 attempts, 4,096-token evaluation limit	115	142
Problems covered at 64 attempts, 8,192-token evaluation limit	142	164
Responses still truncated at 8,192	7,897 (57.1%)	5,409 (39.1%)
Majority vote (%)	40.74	48.15
Budgeted coverage (%)	42.13	50.00
Mean responses within the budget	4.2	6.6
Problems without a finished answer	106	49

The coverage difference is 10.19 ± 4.40 points at 8,192 tokens, against 12.50 ± 4.86 at 4,096; the change between the two limits is -2.31 ± 5.56 points. The majority-vote difference is 7.41 ± 5.10 points.

F Reward deletion

F.1 Intervention

A deletion run still samples the selected prompts and generates their responses. It sets every reward in a selected group to zero before computing advantages, making that group's explicit advantage coefficients zero. It does not remove the prompt from the training set. Subsequent generated responses may differ because the models learn differently.

We use N for the model trained with all rewards, $-\mathrm{H}$ and $-\mathrm{M}$ for hard- and medium-cohort deletion, and $-\mathrm{H}-\mathrm{M}$ for both. R is a random control for the hard cohort; R-M is a uniformly drawn control for the medium cohort; RMS additionally matches the removed absolute response advantage within each data source. These matches are computed on the training records of N, not on the later responses of the changed runs.

We asked whether deleting the hard cohort's own credit improves that cohort's later coverage relative to N and to matched random deletion. It does not. On the held-out suite, hard deletion reduces coverage from 78 to 62 problems, and its difference from the matched random deletion is -2.78 ± 4.63 points.

Table 14: All fixed-count deletion endpoints on 216 problems at the 4,096-token evaluation limit. Each trained row is one realization. Counts and pass@1 are answer-checker metrics. RMS-C is the coefficient-matched control.

Model	Condition	Covered / 216	pass@1 (%)	Mean tokens	At limit (%)
SmolLM3-3B	Base	43	1.02	153	0.09
	N	78	6.05	1,871	13.84
	−H	62	4.28	951	1.12
	−M	59	4.42	3,411	71.36
	R	68	4.40	1,163	2.41
	R-M	65	5.72	2,181	29.96
	RMS	54	4.91	3,586	78.08
	−H − M	53	2.03	371	0.59
Qwen3.5-9B	Base	82	3.31	377	3.31
	N	115	23.39	3,333	70.15
	−M	126	21.77	2,955	60.03
	R-M	115	23.70	3,187	66.02
	RMS	126	23.63	3,203	66.07
	RMS-C	122	24.13	3,185	65.37
Qwen3.5-35B-A3B	Base	111	6.25	674	5.82
	N	137	31.58	2,958	56.63
	−M	134	30.24	2,988	57.45
	R-M	144	30.71	2,869	54.69
	RMS	133	29.00	3,007	59.63

G Fixed-attempt and token-budgeted evaluation

G.1 All deletion checkpoints

Table 15 gives the budgeted results of all deletion checkpoints on their 64 saved responses per problem. The 4,096-token Qwen3.5-9B model N therefore appears with the range 34.72 to 36.11, before the additional responses of Appendix E determined it at 36.11. A coverage outcome can be determined while the vote is not: one correct response settles budgeted coverage, but further responses could still change the majority answer.

On 35B-A3B, R-M covers 144 problems among 64 attempts and N covers 137. The point difference favors R-M, but its interval includes zero. Under the token budget, R-M's majority vote is correct on between three and 12 fewer problems than N: the paired difference lies between -5.56 and -1.39 points. The difference between their budgeted-coverage contrast under the token rule and their 64-attempt coverage contrast is -6.02 ± 5.32 points. Voting over the original 64 attempts, without a token limit, gives both 119 correct answers. The budget therefore changes the comparison, and a majority vote over a fixed number of attempts would not reveal it.

On 9B, medium deletion and RMS each cover 15 problems not covered by N among the original 64 attempts. In each of these two models, only one of those 15 problems has a correct response within the 16,384-token budget. For 35B R-M, none of the 14 problems that only it covers at 64 attempts has a correct response within the budget. Higher coverage at a fixed number of attempts can therefore come from correct answers outside the budget.

G.2 SmolLM3-3B base model with 256 responses per problem

The base model's 64 saved responses run out before the budget on many problems. We therefore combine four samples of 64 responses into 256 responses per problem. The trained model keeps its 64 responses, which suffice for its longer responses. Under the 16,384-token budget, the base model has answer-correct responses on 57 problems and the trained model on 39 (Figure 4).

At 16,384, the base-minus-N difference in budgeted coverage is 8.33 ± 6.48 points. At 4,096 the lower interval endpoint is zero, and at 32,768 the base model's saved responses run out on 44 problems. The base model's lead is therefore clearest at 8,192 and 16,384 tokens. With the same 256 responses,

Table 15: Budgeted results of all deletion checkpoints (percentages). S_{64} votes over 64 attempts without a token budget, and C_B and S_B use the 16,384-token budget. Every model has 64 saved responses per problem. Ranges cover the possible outcomes where these run out and are not confidence intervals.

Model	Condition	S_{64}	C_B	S_B	Undetermined votes
SmolLM3-3B	Base	1.85	19.91 to 94.44	0.00 to 93.52	202
	N	12.50	18.06	8.33	0
	−H	6.02	17.13	5.56 to 6.02	1
	−M	12.96	12.50	9.72	0
	R	6.94	17.59 to 18.06	5.09 to 6.48	3
	R-M	10.19	15.74	6.94 to 7.87	2
	RMS	15.28	11.11	8.80	0
	−H − M	4.17	20.83 to 38.89	3.24 to 25.93	49
Qwen3.5-9B	Base	6.94	32.87 to 45.83	4.17 to 32.87	62
	N	46.76	37.04	34.72 to 36.11	3
	−M	46.30	38.43	32.87 to 35.19	5
	R-M	45.83	36.11	31.94 to 33.80	4
	RMS	49.54	37.50	33.80 to 35.65	4
Qwen3.5-35B-A3B	Base	17.59	42.13 to 44.44	10.19 to 20.83	23
	N	55.09	46.76	42.59 to 44.44	4
	−M	52.31	43.06	38.43 to 40.28	4
	R-M	55.09	43.98	38.89 to 41.20	5
	RMS	54.17	45.37	39.35 to 41.67	5

Table 16: Budgeted coverage of the SmolLM3-3B models at several budgets. The first three rows support the paired coverage comparison. At larger budgets the saved responses run out, and the ranges are not confidence intervals.

Budget	Base coverage (%)	N coverage (%)	Median base attempts	Median N attempts
4,096	13.89	8.80	29	2
8,192	19.44	12.04	58	4
16,384	26.39	18.06	113	9
32,768	31.94 to 52.31	25.46	228	19
65,536	33.80 to 96.76	30.09 to 35.19	256	37
131,072	33.80 to 100	34.26 to 78.24	256	64

every majority vote is determined: the majority answer is correct on 5 problems for the base model and on 18 for N, a difference of 6.02 ± 3.71 points in favor of N (Section 6). The base model's lead therefore concerns budgeted coverage, not the answers the models return.

H Automated audit of written derivations

H.1 Judging procedure and calibration

The audit examines every answer-checker-correct response within the 16,384-token budget. The SmolLM3 base uses its 256 saved responses per problem, and the other models use 64. Muse Spark 1.3 receives only the problem and response, with model identity, condition, problem identifier, draw index, reference answer, and grader score withheld. Chat-template and think-tag markers are stripped; length and writing style remain visible. Every call uses the same rubric.

A response is SUPPORTED when it contains a valid derivation sufficient for its stated answer; a harmless slip is allowed. UNSUPPORTED means that an answer is given without sufficient derivation or a necessary step is only asserted. INVALID means an error affects the conclusion. CANNOT JUDGE records inability to determine the status. A short valid derivation can be supported; abandoned failed attempts followed by a valid derivation are not automatically disqualifying. These labels concern written support, not latent reasoning ability.

Calibration includes four short valid derivations and eight negatives (answer-only responses or deliberately corrupted derivations). All four positives and none of eight negatives are judged supported. The stop condition would trigger if more than one negative were labeled supported. On 24 natural SmolLM3 responses annotated by another model, agreement is 20/24 on the four-way label and 22/24 on supported versus not. These are small calibration checks against model annotations, not human validation or an estimate of judge reliability on all responses.

Table 17: Response-level automated labels. Only answer-checker-correct responses are judged. Response counts are not problem coverage; many responses can concern the same problem.

Model	Condition	Judged	Supported	Unsupported	Invalid	Cannot judge
SmolLM3	Base	331	66	64	201	0
	N	222	157	19	46	0
Qwen 9B	Base	393	191	52	150	0
	N	786	750	26	10	0
Qwen 35B	Base	535	324	61	149	1
	N	1,155	1,078	58	19	0

H.2 Problem-level results

A problem has supported coverage only when at least one of its answer-correct responses is judged supported. The SmolLM3 base's 66 supported responses concern only three problems; 59 of them are concentrated on a single dice problem. N has 157 supported responses on 16 problems. The supported sets overlap on three, with 13 additional N-only problems and none base-only. Their trained-minus-base difference is 6.02 ± 3.01 points, with labels held fixed. This is why the response-level supported fraction is not an adequate substitute for coverage.

Table 18: Problem-level audit results out of 216. Qwen base ranges cover all possible outcomes where the saved responses run out. They are not 95% intervals. Trained counts are observed with the reported labels; disputed labels and judge variation are separate uncertainties.

Model	Condition	Answer-only covered	Judge-supported covered
SmolLM3	Base	57	3
	N	39	16
Qwen 9B	Base	71 to 99	37 to 82
	N	80	74
Qwen 35B	Base	91 to 96	51 to 61
	N	101	94

On Qwen3.5-9B, the base model's saved responses run out before the budget on 62 problems, and 45 of these have no supported response yet. Allowing each unresolved case a supported response gives the upper bound $37 + 45 = 82$, which exceeds the trained count 74. The full-budget ordering is not identified. On 35B, the corresponding base supported bound is 51–61 versus N's 94, so the order holds over the whole range, conditional on the labels. Some coverage outcomes are already determined by an observed correct response even when supported coverage is not, explaining why the widths of the two bounds can differ.

Paired bootstrap intervals that treat the saved Qwen responses as complete would ignore the problems where they run out, so we report counts and ranges instead.

H.3 Independent model checks and remaining limitations

Independent model-based rereadings inspect sampled responses, labels, blinding, and code. The reported checks agree on 12/12 sampled base invalid responses and 8/8 trained supported responses in each family. For trained invalid responses, agreement is 4/4 on SmolLM3 and 3/4 on Qwen. These small, selected samples do not establish an error-free audit.

The Qwen inspection raises substantive disagreements. Five of the 35B problems with a correct but unsupported base response and a supported trained response are disputable on close reading. Some

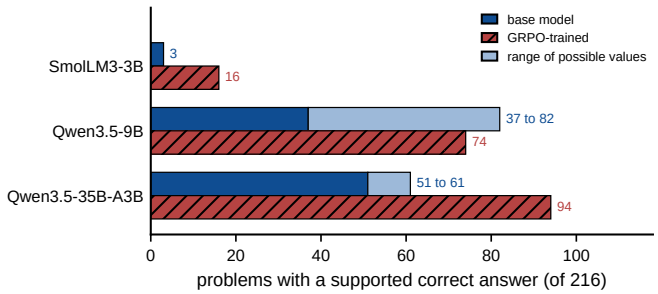


Figure 6: Problems with a supported correct answer within the budget, base against GRPO-trained model. Where a base model's saved responses run out before the budget, the lighter segment spans the values over all possible outcomes. The 9B order is not identified (37 to 82 against 74); the 35B order holds over the whole range, conditional on the judge's labels.

multiple-answer responses also expose a mismatch between the answer discussed by the judge and the answer accepted by the exact-match checker. We retain the reported labels, disclose the issue, and do not present the model checks as a human gold standard or as resolving all disputed cases.

Style can affect judgment: the median response lengths in the 9B audit are 62 words for the base and 230 for N; for 35B, 74 and 224. Almost all Qwen responses contain an "Answer:" line, whereas SmolLM3's trained responses use boxed answers far more often. Short-answer calibration reduces one concern but cannot eliminate length or format bias on natural data. One judge, one pass, and fixed-label bootstrap intervals do not measure this source of uncertainty.

Finally, the audit covers only responses accepted by the answer checker. It does not detect correct derivations that the checker rejected, and it cannot estimate the quality of all generated responses. For the training-limit comparison of Section 5, the same judge labels every correct answer within the budget of the two Qwen3.5-9B runs: 786 answers of the 4,096-token run and 1,123 of the 2,048-token run. The 4,096-token run's labels come from an earlier run of the judge with the same frozen prompt. Re-judging 150 of them in the new run agrees on 146 of 150 supported-or-not labels and changes no problem's status. The 2,048-token run has a supported answer on 94 problems and the 4,096-token run on 74, a difference of 9.26 ± 4.86 points. Of the 26 problems with a supported answer only from the 2,048-token run, the 4,096-token run has no correct answer within the budget on 23 and correct but unsupported answers on 3. An independent LLM re-check of these 3 problems confirms the 2,048-token run's supported answers on all of them, but accepts the 4,096-token run's answers on one of them under a more lenient standard of proof. The difference is then 8.80 points. Of the correct answers, 95 (8.5%) of the 2,048-token run and 36 (4.6%) of the 4,096-token run are not supported, a difference of 3.9 ± 2.5 points with problems resampled.