FROM COMPLETED LENGTH TO PREFIX DECISIONS: A PAIRED BUDGET AUDIT OF REASONING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

A completed response's length can predict correctness while providing little guidance about when to stop generation. We examine this distinction in a paired prefix-replay study of Qwen3-4B-Thinking-2507 and DeepSeek-R1-Distill-Qwen-7B: four sampled streams for each of 300 GSM8K test questions, totaling 2,400 trajectories. An agreement rule stops once two visible numerical answers match; its threshold is chosen on a separate training-split development set. Under a 4,096-token ceiling, agreement scores 58.0% and 59.3%, exactly matching round-robin's question-level correctness. It consumes 131 and 235 fewer tokens per question, but random allocation scores 67.7% and 67.0%. A post hoc control that finishes streams in fixed order reaches 88.3% on both checkpoints. In contrast, retrospective within-question comparisons show longer incorrect than correct responses on the subsets containing both. The results separate that completed-length association from the effect of a prefix-only decision: how the budget reaches a final answer matters more here than stopping on agreement. We report paired uncertainty, all controls, and separate costs for replay and corpus generation. The measured costs count tokens; online serving latency is not evaluated.

1 INTRODUCTION

Reasoning models produce responses whose lengths vary substantially across problems. Longer generation can permit additional calculation, revision, or verification, but it also consumes inference resources. Test-time scaling therefore raises two related questions: does the length of a completed response indicate correctness, and can information available during generation guide a better allocation of the remaining budget? Predicting correctness from a finished response and intervening before it finishes require different information. A correlation between final length and errors establishes only the former.

This distinction is easy to miss when complete trajectories are already available. Their final lengths and final answers can be used to decide retrospectively which responses would have been worth finishing. An online policy cannot access either quantity before the corresponding text has been generated. Evaluating a policy with those future features gives it information that a deployed system would not have. Truncating saved text also creates an outcome that differs from selecting among completed answers: a useful solution can lose its conclusion before a parser can read it.

We study allocation among four response streams under a shared token budget. Four independently sampled response streams are available for each question. A scheduler exposes them in blocks of up to 128 generated tokens under a shared output-token ceiling. A prefix-only rule closes a stream when an explicit final answer becomes available and stops all generation after two observed answers agree. We compare it with uniform per-stream caps, randomized allocation, and round-robin allocation that can reuse an ended stream's unspent share. All policies see prefixes of the same sampled streams. No policy reads a reference answer when choosing which stream to advance or when to stop.

Two checkpoints, Qwen3-4B-Thinking-2507 and DeepSeek-R1-Distill-Qwen-7B, are evaluated on the same 300 GSM8K test questions. Their four streams produce 2,400 confirmation trajectories. A separate training-split development set is used to select the agreement threshold. The analysis reports correctness, answer availability, consumed prefixes, and prompt-token use. The full cost of generating the replay corpus is kept separate from the hypothetical cost of exposing a prefix under each policy.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The paired design tests whether answer-aware stopping adds value beyond budget allocation alone. Agreement and round-robin give identical question-level correctness, whereas a post hoc sequential schedule performs substantially better. The comparison also traces these differences to when each schedule exposes a usable answer. Together with the retrospective length analysis, these results show why a useful association in completed text need not yield a useful stopping rule. The evaluated policy class is fixed-stream replay; learned allocators and online serving systems are outside the comparison.

2 Related Work

Generating and selecting reasoning paths. Chain-of-thought and zero-shot reasoning prompts elicit intermediate work (Wei et al., 2022; Kojima et al., 2022). Self-consistency aggregates sampled solutions (Wang et al., 2023), while verifiers can rank or supervise mathematical reasoning (Cobbe et al., 2021; Lightman et al., 2024; Wang et al., 2024). Tree of Thoughts and Self-Refine introduce search or revision rather than simply reading a fixed set of independent paths (Yao et al., 2023; Madaan et al., 2023). We use voting only among answers already visible at the scheduler's current prefix. This restricts selection to numerical votes available in the current prefixes, without learned verification or revision.

Allocating test-time computation. Snell et al. (2025) study adaptive test-time computation and its dependence on problem difficulty. Budget forcing in s1 changes how long a model continues reasoning (Muennighoff et al., 2025). These methods motivate measuring accuracy jointly with cost, but their interventions differ from ours. Prefix replay keeps each stream's sampled continuation fixed. It can emulate deciding whether to request another block from that stream; it cannot emulate inserting a new instruction, changing the model's context, or resampling after a partial answer. We therefore do not interpret a replay gain as evidence about every way of shortening or extending reasoning.

Uncertainty and evaluation. Calibration evaluates the relation between reported confidence and observed correctness (Guo et al., 2017); semantic uncertainty and self-consistency-based hallucination checks use variation between generated texts or meanings (Kuhn et al., 2023; Farquhar et al., 2024; Manakul et al., 2023). Our stopping signal is much simpler: repeated exact numerical answers after a closing reasoning delimiter. It is neither a calibrated probability nor proof that the agreed answer is correct. Behavioral testing and mathematical robustness studies also show why a single aggregate score can hide important changes in task or measurement (Ribeiro et al., 2020; Li et al., 2024; Mirzadeh et al., 2025). We retain paired question-level outcomes and use the question, rather than each response, as the resampling unit (Dror et al., 2018; Efron, 1979).

3 Completed Length and Prefix Information

3.1 Two different estimands

Let X be a problem, $T = (t_1, \ldots, t_L)$ a completed response, and $Y \in \{0, 1\}$ its scored correctness. A retrospective analysis can estimate a function such as $\Pr(Y = 1 \mid L = \ell)$ or compare the lengths of correct and incorrect responses. These quantities describe completed responses under the collection protocol. They do not identify the correctness of an alternative response generated under a smaller budget.

A scheduler at time u instead has an information set $\mathcal{F}_u$ containing only the problem, the already exposed tokens, and termination events already observed. Its action must be measurable with respect to $\mathcal{F}_u$. The final length L is generally not in this information set. Even when a prefix looks unproductive, the experiment must determine whether stopping it and reallocating the budget yields a better scored answer.

A simple example separates the two questions. Suppose easy problems usually finish in 100 tokens and are correct with probability 0.9, while difficult problems usually finish in 1,000 tokens and are correct with probability 0.5. Length is negatively associated with correctness. Yet a 100-token cap can remove every difficult problem's final answer. Under a final-answer score, that intervention reduces difficult-problem accuracy to zero rather than improving it. The example is an identification

argument, not a fitted model of our data: latent difficulty can generate the association without making shorter reasoning beneficial.

3.2 Answers available during generation

A stream provides a usable answer when its exposed prefix contains a closing `</think>` delimiter followed by a boxed numerical answer. The parser reads only the text after the delimiter. The extra delimiter requirement avoids treating a provisional boxed number inside a reasoning segment as a finished answer. It also defines a restricted interface: a response without the delimiter remains unavailable even if a reader could infer its intended answer.

This policy-specific answer contract differs from simply scoring the last number in a complete response. We state it explicitly because a prefix scheduler can otherwise appear to benefit from intermediate guesses that were later revised. All controls use the same prefix answer contract. The reference answer is used only after the scheduler's decision, to score its selected prediction.

The extraction grammar accepts signed integers and finite decimal numerals, including thousands separators. Equivalent decimal forms are normalized before counting agreement. Thus `16`, `16.0`, and `16.00` represent one vote. Numerical normalization was corrected during collection; the development-selected threshold and generation protocol were retained. Results under the original string semantics are reported as a sensitivity, so the correction is not described as a prospectively frozen analysis choice.

4 The Replay Decision Problem

4.1 Streams, actions, and token ceilings

For question i, the corpus contains streams T_{ij} with sample indices $j \in \{1, 2, 3, 4\}$. Their full lengths are $L_{ij} \leq 4096$. At a replay state, s_{ij} tokens have been exposed from stream j, and total output cost is $C_i = \sum_j s_{ij}$. The scheduler advances one eligible stream by up to 128 tokens, stopping earlier if that stream ends. The total budget is $B \in \{2048, 4096\}$. No block is requested when fewer than 128 tokens remain, irrespective of the unobserved continuation length.

The replay stores decoded prefixes at every 128-token boundary. At a stream's end, later stored boundaries repeat the ended sequence; reading that boundary does not reveal additional tokens. The scheduler charges only tokens actually exposed. The full future length is used by the simulator to implement the environment's termination, not supplied as a scheduling feature. A policy sees that a stream has ended only after advancing it to that endpoint.

Because the decision granularity is 128 tokens, a policy can leave a small remainder of the global budget unused. Uniform per-stream allocation can leave larger unused shares when a stream ends early. These are properties of the allocation rules, not violations of their common budget ceiling. We report realized consumption instead of claiming every policy used exactly B tokens.

4.2 Four allocation rules

Table 1 separates answer-aware stopping from scheduling. Uniform allocation is a common simple control, but its lack of share reallocation can be a substantial restriction. Random allocation can advance some streams more quickly while keeping the same total ceiling. Round-robin uses no answer-aware stopping but removes the fixed-share restriction. Comparing agreement stopping only with uniform allocation would combine both distinctions.

At the final replay state, the prediction is the most common available answer. Missing answers do not vote; if no answer is available, the question is wrong. Vote ties are broken by the earliest sample index among tied answers. This deterministic rule is shared by all policies and does not consult the label. An early agreement can be wrong, so the stopping event itself is not interpreted as a correctness certificate.

Table 1: Decision rules compared on identical sampled streams. Every rule uses the same prefix parser and final voting rule. The extra round-robin control was added during collection, before complete confirmation analysis.

Rule	Next stream	Stream or global stopping
Uniform	Least exposed eligible stream, ties by sample index	Each stream is capped at $B/4$; ended shares are not reused
Random	Uniform random choice among non-ended streams	Ended streams close; remaining streams may reuse the budget
Round-robin	Least exposed non-ended stream, ties by sample index	Ended streams close; unused shares may be reallocated
Agreement	Least exposed eligible stream, ties by sample index	A visible answer closes its stream; two equal visible answers stop the question

4.3 What paired replay identifies

Conditional on the saved stream corpus, every policy has a deterministic outcome except for the seeded random allocator. This design removes variation from resampling a fresh set of answers for each policy. The paired difference therefore measures how allocation changes the use of those particular streams. It does not include effects from policy-dependent resampling, changes in prompt text, or different runtime scheduling implementations.

There is a plausible online interpretation when a system can pause and resume independent autoregressive streams without altering their conditional distributions. Under that assumption, requesting another block reveals the same kind of continuation simulated here. We do not implement or benchmark such a server. In particular, batching, cache management, and context-dependent attention costs are not measured by counting output tokens. A measured prefix-token saving is an accounting result rather than a measured wall-clock speedup.

5 Experimental Protocol

5.1 Data, models, and generation

We use GSM8K (Cobbe et al., 2021). A deterministic hash of question indices selects 100 training questions for development and 300 test questions for confirmation. The development and confirmation sets are disjoint by dataset split. Both models use the same questions and four sample indices per question. Qwen3-4B-Thinking-2507 (Yang et al., 2025; Qwen Team, 2025) and DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025) are treated as two distinct checkpoints, not as a controlled model-size or distillation comparison.

Each input uses the checkpoint's chat template and asks the model to reason through the problem and place the final numerical answer in a box. Responses use temperature 0.7, top-p 0.95, and a ceiling of 4,096 new tokens. The sample seed is a deterministic function of the question and sample indices. We retain all complete text, generated token IDs, prefix text, prompt-token counts, sample IDs, and termination status. Models use bfloat16 and the same stated attention backend; generation order can differ across execution shards without changing the assigned questions, sample indices, or seeds.

The full collection contains 800 development trajectories and 2,400 confirmation trajectories. The confirmation results do not reuse training-split questions to select a favorable threshold. Exact question identities and sample counts are checked before aggregate analysis, and every selected question contributes all four streams per model. Missing answers remain in the denominator.

5.2 Development choice and analysis amendments

On the development data, agreement thresholds two and three are compared across both models and both token ceilings. The threshold is selected by mean development accuracy, with lower consumed cost breaking a tie. This rule selects two. The selected threshold is then fixed for the confirmation analysis. All planned controls remain in the results.

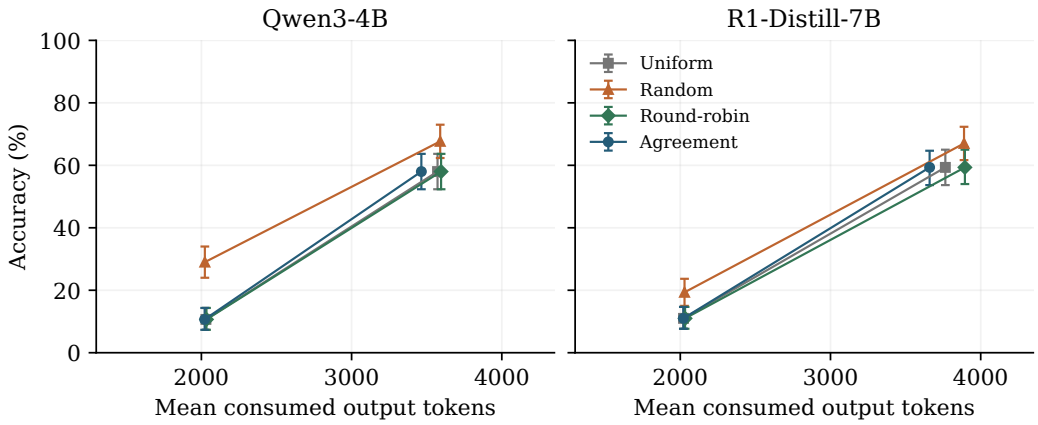

Figure 1: Accuracy and realized output-token consumption under the two evaluated ceilings. Lines connect the 2,048- and 4,096-token evaluations; they are not fitted scaling curves. Error bars are descriptive 95% question-bootstrap intervals. The four main rules use the same sampled streams within a model.

Two changes are distinguished from that frozen choice. Equivalent numeric labels are normalized consistently in voting and scoring, and a round-robin control is added to isolate redistribution of unused shares. Both changes are documented as analysis amendments made during collection, not as parts of the original threshold-selection protocol. The original string-semantic outcomes are retained. A post-collection audit additionally makes the final-block decision causal: fewer than 128 remaining tokens always stops replay, rather than consulting the unseen terminal length. On every confirmation question, this correction preserves scored correctness; it reduces mean cost by at most 2.23 tokens per condition. Both implementations are supplied. None of these changes modifies generated text or reference labels.

5.3 Endpoints and uncertainty

For each model, budget, and policy, we report accuracy, mean consumed output tokens, mean prompt tokens of activated streams, and the number of activated streams. The prompt cost counts a stream's input when it is first activated. This is a transparent accounting convention, not a claim about a runtime's prefix-sharing implementation. Total source-generation tokens are reported separately and are not subtracted as hypothetical savings.

The primary comparisons are paired question-level accuracy differences between agreement and each simple control. We use 20,000 seeded bootstrap resamples of the 300 questions, carrying all four streams and all policies together within each resampled unit. Intervals are descriptive and are not adjusted to provide simultaneous coverage over every model, budget, and endpoint. We do not treat four streams from one question as four independent evaluation questions or interpret an interval spanning zero as evidence of equivalence.

We also examine the relation between complete-response length and correctness. That analysis uses only finished corpus records and is labelled retrospective. It does not contribute future features to the scheduler. Comparing it with the prefix results is the purpose of the experiment: a descriptive length signal and an actionable budget rule can lead to different conclusions.

6 Results

6.1 Accuracy under common token ceilings

Table 2 reports the four main policies at both ceilings. The ceiling controls the maximum permitted output, while the plotted horizontal position in Figure 1 shows realized consumption. These quantities differ when paths end or agreement stops the question early. For Qwen3-4B at 4,096 tokens,

Table 2: Confirmation accuracy (%) on 300 questions per checkpoint. All four streams are available to every policy, subject to its scheduling and stopping rules.

Checkpoint	Ceiling	Uniform	Random	Round-robin	Agreement
Qwen3-4B	2,048	10.7	29.0	10.7	10.7
Qwen3-4B	4,096	58.0	67.7	58.0	58.0
R1-Distill-7B	2,048	11.0	19.3	11.0	11.0
R1-Distill-7B	4,096	59.3	67.0	59.3	59.3

Table 3: Mean consumed output tokens at the 4,096-token ceiling and total tokens used to generate each model's 1,200 confirmation streams. Corpus generation is a separate experimental cost, not a saving credited to a replay policy.

Checkpoint	Uniform	Random	Round-robin	Agreement	Corpus tokens
Qwen3-4B	3,572	3,590	3,596	3,464	1,826,507
R1-Distill-7B	3,765	3,889	3,895	3,660	1,852,497

agreement minus round-robin is 0.0 percentage points, with a descriptive 95% interval of [0.0, 0.0]. Agreement minus random allocation is -9.7 points, with interval [-13.3, -6.3]. For R1-Distill-7B at 4,096 tokens, agreement minus round-robin is 0.0 percentage points, with a descriptive 95% interval of [0.0, 0.0]. Agreement minus random allocation is -7.7 points, with interval [-11.7, -3.7]. The intervals favor random allocation over agreement on both checkpoints. The zero-width round-robin contrast reflects identical correctness indicators in this corpus, not population equivalence.

6.2 Replay consumption and corpus-generation cost

Table 3 keeps the two uses of compute distinct. Replay consumption is the amount of saved text a policy would expose. Source generation produced all four full streams irrespective of which policy was later evaluated. The accompanying data also retain activated-stream and prompt-token counts. We do not subtract the cost of an unsuccessful or unparsed stream from either total.

6.3 Retrospective length evidence

Figure 2 uses complete responses, including those that do not produce a scored final answer. For a paired descriptive check, we compute each question's mean incorrect-response length minus its mean correct-response length when both types occur among its four samples. This conditional set can differ between models. For Qwen3-4B, 39 questions meet that condition; the mean within-question length difference is 1394 tokens, with interval [1183.0, 1603.3]. For R1-Distill-7B, 42 questions meet that condition; the mean within-question length difference is 1203 tokens, with interval [854.5, 1537.1]. This comparison describes observed complete paths. Neither the final lengths nor their correctness labels are available to the scheduler before generation, so it cannot be interpreted as a deployable correctness predictor or as evidence that stopping an incorrect-looking path would make its answer correct.

6.4 Numerical representation and simple controls

The original string-semantic analysis is retained beside canonical-number scoring. Both versions use the same saved trajectories and scheduler definitions. Numerical normalization can change a vote and the time at which agreement triggers stopping; the resulting differences are analysis effects on fixed generations. The additional round-robin comparison similarly isolates a specific control choice: whether a stream's unused share can be spent elsewhere. Neither sensitivity is treated as a new independent confirmation sample. At 4,096 tokens, Qwen3-4B's agreement accuracy is 57.7% under original string semantics and 58.0% under canonical-number scoring. At 4,096 tokens, R1-Distill-7B's agreement accuracy is 59.3% under original string semantics and 59.3% under canonical-number scoring.

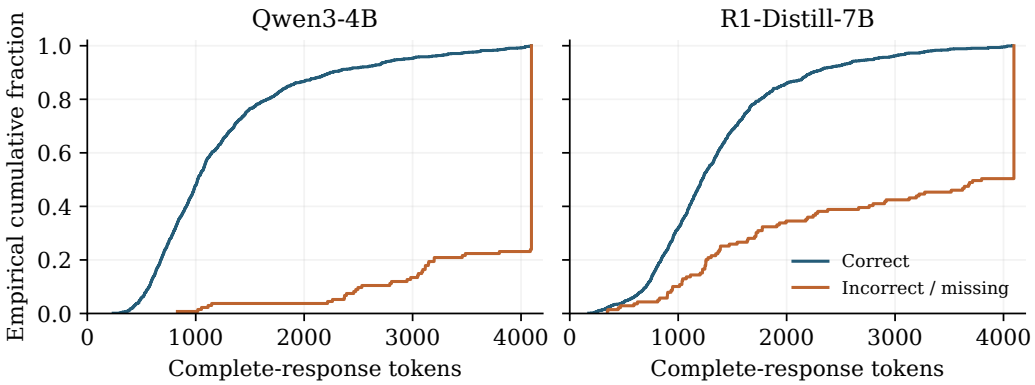

Figure 2: Complete-response length distributions, stratified by scored correctness. Each curve is descriptive; outputs from one question are dependent. A concentration of wrong or missing answers near the generation ceiling does not identify the effect of stopping them earlier.

6.5 A POST HOC SEQUENTIAL CONTROL

An additional control completes streams in fixed sample-index order before advancing to the next one. It uses the same causal 128-token blocks, final vote, and total ceilings, with no answer-aware stopping or gold-dependent selection. This control was added after the main results and is evaluated on the same saved responses, not on new independent data.

At 2,048 tokens, sequential allocation reaches 76.7% accuracy for Qwen and 77.7% for DeepSeek; at 4,096 it reaches 88.3% for both. Mean consumption at the larger ceiling is 3,572 and 3,875 tokens. Agreement minus sequential is -30.3 percentage points for Qwen, with paired 95% interval $[-35.7, -25.0]$, and -29.0 for DeepSeek, with interval $[-34.3, -23.7]$. Thus the weakness of balanced short prefixes is not resolved by comparing only with random allocation: a simpler schedule that lets paths finish is substantially stronger in this corpus. This result limits claims about the evaluated agreement rule, without establishing an optimal general scheduling policy.

6.6 WHERE THE SCHEDULING DIFFERENCE ARISES

A total budget is useful only when it exposes an answer. Under a uniform 4,096-token allocation to four streams, each stream receives at most 1,024 tokens. A question whose four answers all first appear later than that threshold can exhaust the entire budget without exposing a single vote. A random schedule can instead spend more than one quarter of the budget on an individual stream. It may reach an answer even though it has no information about which stream will be correct. The gain can therefore arise from reaching an answer sooner, even when allocation carries no information about its correctness.

Figure 3 examines this mechanism without using correctness labels. At each common per-stream prefix, we count whether at least one or at least two of the four streams contain a visible answer under the same extractor. The second event is necessary, but not sufficient, for a two-answer agreement: the two values must also match. The plot concerns observable answer availability, not the quality of an answer or the effects of changing how it was generated.

Table 4 reports the realized allocation and its exposed votes. These quantities are computed after applying each complete policy; they do not enter the scheduler. Spread describes allocation inequality, not correctness.

The equality of round-robin and agreement accuracies is stronger than a nonsignificant difference: their question-level correctness indicators are identical at the reported ceilings. The resulting paired bootstrap interval is exactly zero because resampling identical differences cannot create variation. It is not a population equivalence certificate. The policies can consume different numbers of tokens or produce different answer sets while agreeing on whether their final vote is correct in this finite corpus.

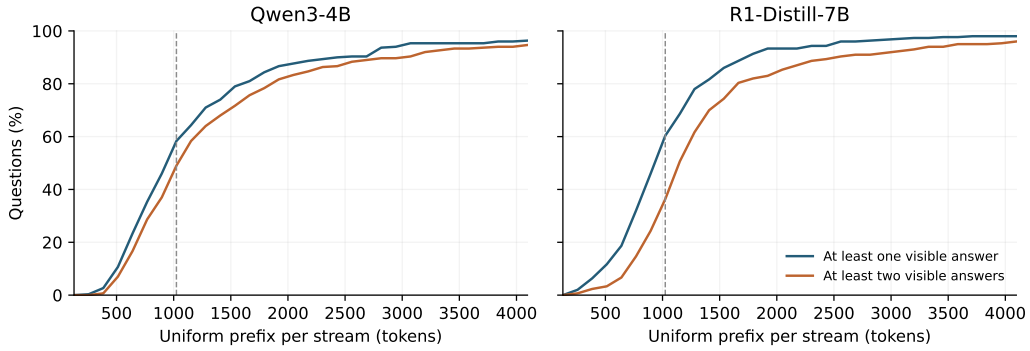

Figure 3: Answer visibility when all four saved streams receive the same prefix length. The dashed line marks 1,024 tokens per stream, the uniform share of the 4,096-token budget. Two visible answers need not agree. This post hoc mechanism diagnostic uses every confirmation question and no correctness-based selection.

Table 4: Scheduling diagnostics at a 4,096-token ceiling. Visible answers are the mean count per question; no answer is the percentage with none; allocation spread is the mean difference between the most- and least-exposed stream, in tokens.

Checkpoint	Policy	Visible answers	No answer (%)	Spread
Qwen3-4B	Uniform	1.82	41.7	102
	Random	2.12	32.0	458
	Round-robin	1.94	41.7	124
	Agreement	1.08	41.7	67
R1-Distill-7B	Uniform	1.21	39.7	172
	Random	1.68	31.7	640
	Round-robin	1.54	39.7	271
	Agreement	1.04	39.7	135

Agreement therefore has an observed token-cost effect here without an observed accuracy benefit over the matched allocation control.

These diagnostics were constructed after examining the complete results to explain the scheduling contrast. They are not additional confirmation tests. They leave the original primary comparison unchanged and cannot establish the performance of a newly designed allocation rule. The post hoc sequential control examines one such alternative on this corpus. Changing chunk size, adapting stream order, or choosing a model-specific threshold would require further evaluation.

6.7 Interpreting the paired comparison

The unit of the comparison is the question, with its four saved responses held fixed. This controls the realized response pool: every policy faces the same available reasoning paths, rather than receiving an independently favorable or unfavorable set of samples. It also makes the conclusion conditional on that pool. A second experiment with newly sampled responses could change the numerical differences even on the same questions. The reported question-bootstrap intervals capture variation among the selected questions, not all uncertainty from sampling seeds or model releases.

The random baseline likewise uses a fixed seeded allocation for each question. We do not average over an independent distribution of scheduler seeds, and the intervals should not be interpreted as such an average. The package distinguishes scheduler and generation seeds; the comparison does not establish an optimal scheduler.

7 Interpretation and Limitations

Answer agreement and budget allocation are separate mechanisms. The four rules isolate several effects that are often combined. Fixed shares restrict how far any one path can progress. Reallocating unused shares changes that restriction without interpreting the answer. Early answer detection can close a stream before its natural end, while agreement can stop all streams. A comparison against the random and round-robin controls is therefore necessary to attribute an effect to answer-aware stopping rather than to a different allocation schedule.

A token budget is not a common compute budget across checkpoints. The two models have different sizes, architectures, and tokenizers. The same output-token ceiling does not make their FLOPs or latency equal. Even within one model, the distribution of work across streams changes context lengths and batching opportunities. Our resource comparisons are within each checkpoint under stated token-accounting rules. We make no model-size scaling claim and no claim that saved prefixes translate one-for-one into deployment speed.

A replay corpus limits the policy class. The policy may choose which stream to advance and when to stop. It cannot ask for a revised explanation, alter a prompt, invoke an external verifier, or use a learned confidence model that has not been evaluated here. The absence of a benefit for one agreement rule would not rule out those alternatives. Conversely, a benefit on this fixed corpus would not certify a generally optimal stopping rule. The corpus supports transparent paired comparisons within this restricted action space.

The answer contract affects both availability and correctness. Requiring a closing reasoning delimiter and a boxed answer is a deliberate guard against reading unfinished reasoning as a decision. It can also omit a mathematically correct conclusion expressed in another form. All policies use the same rule, but their exposed prefixes can reach that form at different times. Availability should therefore be read alongside accuracy. A missing prefix answer is a failure under the evaluated decision contract, not a proof that the full underlying trajectory lacks a correct solution.

Scope of generalization. The study covers two checkpoints, one fixed 300-question subset, two ceilings, and one development-selected threshold. It does not evaluate hidden-state probes, broad confidence calibration, or distillation effects. Question-bootstrap intervals do not capture variation across model families or reasoning domains.

Length associations depend on the selected questions. The paired within-question length statistic is large for questions that contain both correct and incorrect samples, yet the agreement rule does not improve accuracy over round-robin. These findings are compatible. The first statistic uses completed responses and labels to define its comparison; the second rule must act before that information exists. In addition, the conditional length analysis excludes questions on which all four responses have the same correctness. Its selected set contains 39 Qwen questions and 42 DeepSeek questions, rather than all 300. Extrapolating that association to every question would change the population being described. A prospective length-based predictor would need its own observable prefix features, development procedure, and held-out decision evaluation.

8 Conclusion

A retrospective length association did not translate into an effective agreement-stopping rule on this corpus. At the 4,096-token ceiling, agreement and round-robin score 58.0% and 59.3% on the two checkpoints, with identical question-level correctness. Agreement saves 131 and 235 tokens per question, while a post hoc sequential control reaches 88.3% on both. Evaluating a stopping rule therefore requires the prefixes, allocation alternatives, and costs available when the decision is made. Completed-response length alone cannot establish that shorter reasoning would improve an answer.

Reproducibility Statement

The supplementary package contains the fixed question indices, development selection, complete response records, decoded prefixes, generation configuration, replay implementation, and scripts for all tables and figures. Model weights and the public GSM8K data are obtained separately. The scoring and bootstrap seeds are specified, and the package retains the original string-semantic analysis as a sensitivity. A companion answer-concentration audit (Anonymous Authors, 2026) shares 37 questions and 296 response records with this corpus across the two checkpoints. The studies address different endpoints and are not independent replications.

Ethics Statement

This study uses public mathematical questions and public language-model checkpoints. It contains no human-subject experiment or personal data collection. Accuracy and cost conclusions apply to the tested replay protocol and should not be interpreted as guarantees about a deployed reasoning service.

AI Assistance

OpenAI Codex was used to aid and polish manuscript writing and to assist with literature retrieval and discovery, including finding related work. The authors are responsible for the final manuscript and citations.

References

Anonymous Authors. Answer diversity, missing outputs, and sampling budgets in reasoning language models. Companion manuscript, 2026.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The hitchhiker's guide to testing statistical significance in natural language processing. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1383–1392, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1128. URL https://aclanthology.org/P18-1128/.

Bradley Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979. URL https://doi.org/10.1214/aos/1176344552.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 2024. doi: 10.1038/s41586-024-07421-0. URL https://www.nature.com/articles/s41586-024-07421-0.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, 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://www.nature.com/articles/s41586-025-09422-z.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large Language Models are Zero-Shot Reasoners. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. GSM-plus: A comprehensive benchmark for evaluating the robustness of LLMs as mathematical problem solvers. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2961–2984, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.163. URL https://aclanthology.org/2024.acl-long.163/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative Refinement with Self-Feedback. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9004–9017, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.557. URL https://aclanthology.org/2023.emnlp-main.557/.

Iman Mirzadeh, Keivan Alizadeh-Vahid, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 94743–94765, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ec2e7a896f8250986b3907f57621ce94-Paper-Conference.pdf.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Fei-Fei Li, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Qwen Team. Qwen3-4B-Thinking-2507 model card, 2025. URL https://huggingface.co/Qwen/Qwen3-4B-Thinking-2507. Official checkpoint documentation.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442/.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b623663fd9b874366f3ce019fdfdd44-Abstract-Conference.html.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

A Replay Accounting

Let a_{ij} indicate whether stream j is ever activated. Output cost is $\sum_j s_{ij}$, while the input-token accounting endpoint is $\sum_j a_{ij} p_{ij}$ for prompt lengths p_{ij}. These are different quantities. The source corpus generation cost is $\sum_j L_{ij}$ regardless of which prefixes a replay later exposes. The package reports all three rather than treating source generation as free. Early stream termination can leave budget unused; the global ceiling is never exceeded. For the compared budgets, uniform shares are multiples of the 128-token decision block.

B Numerical and Scheduling Sensitivities

The primary numerical representation maps finite decimal values to one canonical answer label. The sensitivity preserves the original decimal string representation. Both use the same generated texts, threshold, budgets, and vote-tie rule. Random allocation is seeded by question index. Round-robin allocates to the least exposed open stream and can redistribute unused shares but never stops because of an answer or agreement event. These controls isolate specific scheduling mechanisms and are not a benchmark of all inference-time search methods.