Extended Abstract

Motivation Reasoning-intensive tasks demand substantial inference compute, and adaptive test-time methods reduce this cost by sampling multiple solutions and stopping early once they agree. These methods are increasingly applied to models fine-tuned with reinforcement learning, yet RL fine-tuning tends to collapse a model's output distribution toward high-reward behaviors, and prior work reports that this can degrade calibration, so a model's confidence no longer tracks its correctness. If RL similarly distorts the agreement signal that adaptive stopping relies on, applying these methods to RL-fine-tuned policies risks halting early on confident but incorrect answers. The project addresses whether training regime changes the reliability of this stopping signal, as opposed to merely the accuracy of the policy.

Method I evaluated self-consistency-based adaptive sampling across three training regimes of the same base model (Qwen2.5-0.5B): supervised fine-tuning (SFT), preference optimization (IPO), and online reinforcement learning (RLOO), all on the Countdown arithmetic task. The stopping rule sampled sequentially and committed to the first answer reaching an agreement threshold M, capped at N samples, and was compared against fixed-k majority voting at matched compute. The framework was fully offline: each checkpoint's completions were generated once and cached, then replayed to compute all metrics and stopping simulations without re-invoking the model. The approach is novel because it runs entirely on cached samples, requiring no reward model or retraining; it links calibration (ECE) to adaptive efficiency as the candidate mechanism connecting training regime to stopping reliability; and it holds the prompt set, sample budget, and stopping rule fixed across regimes, isolating the effect of the training objective.

Implementation All three policies were initialized from a common SFT base. Each was evaluated by generating $K = 64$ samples per prompt on a held-out set of 200 Countdown problems, with SFT swept across six temperatures to trace a calibration reference curve and IPO and RLOO evaluated at temperature 0.7. Adaptive accuracy and sample cost were averaged over 200 random sample orderings per prompt. The headline result is a dissociation between accuracy and calibration: RLOO was the least accurate regime yet the best calibrated, while IPO was most accurate but only second best calibrated, and both fell below the calibration frontier traced by varying SFT's temperature.

Results Training reshapes the agreement–correctness relationship beyond what temperature sharpening alone achieves: IPO and RLOO are better calibrated than any SFT temperature setting, in the direction of improved rather than degraded calibration, contrary to the concern motivating the study. Because RLOO ranks lowest on accuracy but highest on calibration, the reliability of agreement-based stopping is a distinct property from accuracy and is not recoverable from it. The reliability diagrams further show that all regimes are systematically underconfident on Countdown: the plurality answer is correct more often than its agreement level implies, so the dominant error is underconfidence rather than the confident-but-wrong failure the study set out to test, which does not appear.

Discussion The underconfidence is consistent with Countdown's answer structure, in which correct solutions collapse to a single target value while incorrect ones scatter, so the calibration findings may be specific to tasks with this structure. The practical implication is that agreement-based adaptive stopping can be applied to RL-fine-tuned policies without the calibration penalty prior work might predict, but because stopping reliability depends on training regime and task structure rather than accuracy alone, a more accurate model is not automatically safer to stop early on.

Conclusion Training improved the calibration of the agreement signal rather than degrading it: both trained regimes fell below the calibration frontier reachable by sharpening SFT's sampling temperature, and RLOO was the best calibrated despite being the least accurate. Thus, calibration and accuracy ranked the regimes differently, which means the reliability of agreement-based early stopping is a property of the training regime, distinct from raw accuracy and not recoverable from it. Adaptive stopping matched fixed-budget accuracy at lower mean compute without a calibration penalty, so it can be applied to RL-fine-tuned reasoning policies without the degradation prior work might predict.

Adaptive Test-Time Sampling for RL-Fine-Tuned Reasoning Policies

Andrea Nam
Department of Computer Science
Stanford University
anamsong@stanford.edu

Abstract

Self-consistency and agreement-based adaptive sampling can reduce inference cost by drawing more samples on hard problems and stopping early when generated answers agree, but it is unclear whether agreement reliably signals correctness, and whether this depends on how the policy was trained. Prior work suggests reinforcement-learning fine-tuning can degrade calibration. This project evaluates adaptive test-time sampling across three training regimes of the same base model (supervised fine-tuning, preference optimization via IPO, and online RL via RLOO) on the Countdown reasoning task, applying a self-consistency stopping rule and measuring expected calibration error (ECE) between sample agreement and correctness alongside accuracy at matched compute. The experiment reveals that training reshapes the agreement-correctness relationship beyond mechanical sharpening: both IPO and RLOO fall below the calibration frontier traced by varying SFT's sampling temperature, and RLOO is the best-calibrated regime despite being the least accurate, although all regimes remain underconfident on Countdown. Hence, the reliability of agreement-based stopping is distinct from accuracy and not predicted by it, indicating that adaptive stopping can be applied to RL-fine-tuned reasoning policies without a calibration penalty, although agreement remains an imperfect correctness signal.

1 Introduction

Rising compute costs have raised practical and financial concerns for both researchers and practitioners deploying large language models. Reasoning-intensive tasks compound this pressure, since accurate solutions often demand substantially more inference compute than simpler tasks. A growing body of work treats inference budget as a resource to be allocated adaptively rather than spent uniformly: sampling multiple reasoning paths can improve accuracy (Wang et al., 2023), and adaptive methods reduce cost by directing more compute to harder inputs (Aggarwal et al., 2023; Snell et al., 2024). Systems such as OpenAI's o1 series have brought these ideas into practice, improving accuracy by spending more computation on harder problems at the cost of additional inference (Se and Vert, 2025).

These adaptive strategies are increasingly applied to models that have themselves been fine-tuned with reinforcement learning, now a standard step for reasoning models. Yet a tension underlies this combination. Reinforcement-learning fine-tuning tends to collapse a model's output distribution toward high-reward behaviors, and prior work reports that this can degrade calibration, so that a model's confidence no longer tracks its correctness (Sahoo, 2026). If RL training similarly distorts the agreement signal that adaptive stopping relies on, then applying these methods to RL-fine-tuned policies risks halting early on confident but incorrect answers. Despite this, the interaction between adaptive test-time compute and reinforcement-learning-trained reasoning policies remains underexplored.

Stanford CS224R 2025 Final Report

This project investigates adaptive test-time compute allocation for RL-fine-tuned language models on the Countdown math reasoning task, using a self-consistency-based stopping rule that allocates more samples to problems where the model's initial generations disagree. Specifically, the project seeks to answer: do RL-trained policies (IPO, RLOO) interact differently with adaptive test-time compute than SFT-only policies? Critically, the value of adaptive stopping depends not on a policy's raw accuracy but on whether sample agreement reliably signals correctness. A more accurate policy could still be unsafe to stop early on if its agreement decoupled from correctness, so the practical question is whether training regime changes the reliability of the stopping signal, not merely the accuracy of the policy. Therefore, the project characterizes when and how reasoning agents can save compute on problems they solve confidently and redirect it to those requiring more deliberation, and ask whether compute savings come without cost or whether there is a trade-off between efficiency and reliability that depends on how the policy was trained. The central hypothesis is that a policy trained with a reward signal directly tied to task correctness should not only solve more problems but should also be better calibrated: when it generates the same answer repeatedly, that repetition should be a more reliable indicator of correctness than it is for a policy trained only on human preference pairs or next-token likelihood. If this holds, an agreement-based stopping rule fires earlier on correct answers and later on incorrect ones for RL-trained policies, yielding a better accuracy-vs-compute tradeoff.

2 Related Work

Recent work on test-time compute has reframed inference budget as a resource that can be allocated adaptively rather than treated as fixed. Snell et al. show that, under a fixed compute budget, allocating more inference effort to harder inputs can outperform simply scaling model size, with the largest gains on inputs of intermediate difficulty (**?**). Related work on reinforcement-learning-based reasoning further suggests that training can improve not only answer quality but also how effectively a model uses compute at inference time (Qu et al., 2024). Nevertheless, these works either study allocation within a single fixed model or train the allocation policy into the model itself; neither asks whether the training regime changes how a fixed evaluation-time sampling rule behaves. That is the question central here: when adaptive sampling is applied at evaluation time, do RL-fine-tuned reasoning policies behave differently from supervised-only policies, even when the evaluation-time compute rule is held fixed?

A closely related line of work uses confidence or agreement to decide when to stop sampling. Self-consistency samples multiple reasoning chains and aggregates them by majority vote, showing that agreement across samples can be informative for correctness on math reasoning tasks (Wang et al., 2023). Confidence-based early exiting applies a related principle at the token level, allowing easier inputs to terminate computation in earlier layers while harder ones use more (Schuster et al., 2022). However, these methods operate at a different granularity from sample-level stopping, and each typically evaluates a single model or a single training family, so they do not isolate how the training regime changes the stopping signal itself. None of these methods compares supervised fine-tuning, preference optimization, and on-policy RL under the same adaptive sampling rule, nor asks whether the agreement signal remains equally reliable across these checkpoints. This matters because prior work documents that reinforcement-learning fine-tuning can degrade calibration by collapsing output distributions toward high-reward behaviors (Sahoo, 2026; Kadavath et al., 2022). Yet, this has been characterized for token-level probabilities, not the sample-agreement signals that adaptive stopping relies on, leaving open whether the same degradation carries over.

The resulting gap is that previous work has not systematically compared adaptive sampling under matched compute across supervised-only and RL-fine-tuned checkpoints of the same base model. This project fills that gap by evaluating self-consistency-based adaptive sampling on SFT, IPO, and RLOO checkpoints, measuring not only accuracy and compute savings but also how training regime affects the reliability and hyperparameter sensitivity of the agreement-based stopping signal.

3 Method

All three policies (SFT, IPO, and RLOO) are evaluated on the same prompts, with the same sample budget K and the same stopping rule, so that differences in the agreement–calibration–efficiency

tradeoff are attributable to the training objective rather than to generation settings or evaluation protocol.

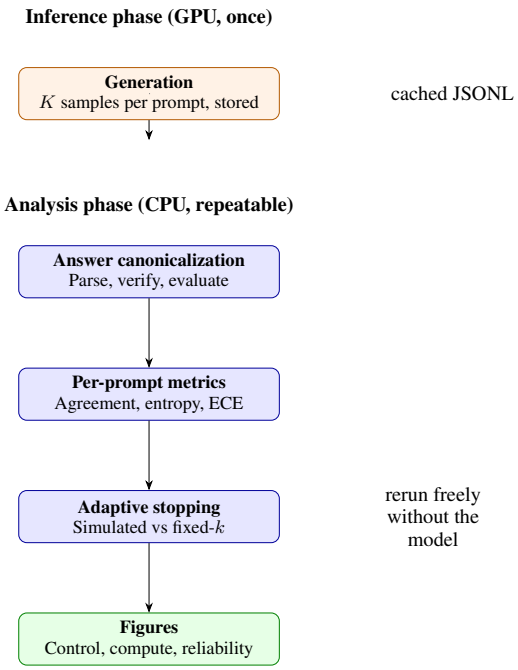

Figure 1: Adaptive sampling pipeline stages.

3.1 Offline simulation design

The pipeline separates a one-time inference phase, which requires a GPU and a running vLLM engine, from a repeatable analysis phase that operates entirely on cached samples. Generation runs once per checkpoint; all metrics, stopping simulations, and figures are then computed from the cache without reloading any model, so the analysis can be iterated freely. For each (checkpoint, temperature) pair, I generated K independent completions per prompt and stored one record per sample, containing the prompt identifier, target value, input numbers, the raw completion, its parsed numeric value, and a correctness flag. Generation is skipped if the cache already exists, making analysis reruns free. The SFT checkpoint was sampled across a grid of temperatures so it can serve as a control curve; IPO and RLOO were each evaluated at a single temperature.

3.2 Answer canonicalization

Each completion was parsed in three stages, reusing the Countdown verifier so that the evaluation definition of correctness is identical to the reward definition used in training. First, the answer expression was extracted from the model's `<answer>` tag; if no valid tag was present, the sample was treated as an abstention. Second, the equation was checked to use exactly the provided input numbers, with no substitutions, extras, or omissions. Third, the expression was evaluated by a restricted arithmetic parser that permits only numeric constants and the four basic operations with unary signs, mapping division by zero and non-finite results to a null value; untrusted model output is never passed to a general evaluator. A sample failing any stage was recorded as an abstention with no parsed value. Abstentions cast no votes but count toward K in all denominators. This is intentional: a model that often fails to produce a parseable answer is genuinely less informative per sample, and removing abstentions from the denominator would artificially inflate apparent confidence.

3.3 Per-prompt metrics

From each prompt's K samples I derive four quantities. Agreement fraction is the share of all K samples, including abstentions, whose parsed value equals the plurality answer; it is the primary

confidence proxy. Under i.i.d. sampling from the model's conditional distribution, this fraction is an empirical estimate of the probability the model assigns to its top answer, and if that answer is correct, of its probability of correctness on the prompt. Plurality correctness is a binary label indicating whether the top answer equals the target. Shannon entropy is computed over the full answer distribution, with abstentions treated as a dedicated outcome rather than dropped, so a model producing mostly unparseable output registers high entropy even if its few valid answers agree. Expected Calibration Error (ECE) Lee et al. (2024) bins prompts by agreement fraction into ten equal-width intervals on $[0, 1]$ and takes the weighted mean absolute gap between mean confidence and mean accuracy per bin:

$$\mathrm{ECE} = \sum_{b=1}^{10} \frac{|B_b|}{N_p} \left| \bar{\mathrm{acc}}_b - \bar{\mathrm{conf}}_b \right|,$$

where $\bar{\mathrm{conf}}_b$ is the mean agreement fraction and $\bar{\mathrm{acc}}_b$ the mean plurality correctness of prompts in bin b. A low ECE means that when the model produces the same answer k out of K times, the fraction k/K accurately predicts correctness, which is precisely the property that makes agreement-based stopping trustworthy.

3.4 Adaptive stopping simulation

The stopping rule is simulated from the cached samples; no model is called. For a threshold M and cap N, the samples of a prompt are processed one at a time in a random order, maintaining a running count for each distinct parsed value, and the rule commits to the first value reaching M votes. If the cap is reached first, the plurality of observed values is taken.

```
ADAPTIVE-STOP(samples, M, N, order):
    counts <- empty map
    for i = 1 .. min(N, |order|):
        v <- samples[order[i]].parsed_value
        if v is not None:
            counts[v] <- counts[v] + 1
            if counts[v] >= M:
                return (v, correct[v], used=i, early=True)
    if counts non-empty:
        best <- argmax_v counts[v]
        return (best, correct[best], used=N, early=False)
    return (None, False, used=N, early=False)
```

The threshold M controls the confidence–cost tradeoff: a low M commits quickly on weaker evidence, while a higher M waits for stronger consensus at greater sample cost. Because the cached samples are i.i.d. and carry no meaningful order, the rule's accuracy and mean cost depend on the arrival order; I estimate their expected values by averaging over many independent random orderings per prompt, with each prompt's seed fixed deterministically so the experiment is reproducible from a single global seed. Dataset-level accuracy and mean samples used are the averages of these per-prompt estimates. The threshold M is swept and the cap is set to $N = K$, giving several adaptive operating points per checkpoint. The fixed-k baseline takes the plurality of the first k samples, tracing the accuracy-vs-compute frontier under no early stopping.

The algorithm was designed to output three the summarized plots. The control plot places mean agreement fraction against ECE, with SFT forming a temperature-parameterized curve and IPO and RLOO as single points. Since temperature trades diversity against agreement, the SFT curve gives the baseline confidence–calibration relationship reachable by sampling alone; a policy whose point falls below the curve (lower ECE at similar agreement) has confidence more reliable than temperature alone explains. The accuracy-vs-compute plot overlays fixed-k curves and adaptive operating points per checkpoint, visualizing whether adaptive stopping is Pareto-efficient: a point above and to the left of the fixed-k curve achieves similar accuracy with fewer samples. The reliability diagrams show per-bin accuracy against the perfect-calibration diagonal for each condition, identifying whether overconfidence (accuracy below the diagonal) or underconfidence (above) dominates.

Prior test-time-compute methods typically train models to support adaptive computation, for example via process reward models or step-level value functions, operate online with the model in the loop,

or evaluate adaptive strategies only on models never fine-tuned with a correctness reward. This extension differs in three ways. First, the simulation is entirely post-hoc over pre-generated samples, requiring no reward model, no retraining, and no change to generation, making it a zero-cost analysis layer applicable to any checkpoint. Second, it connects the ECE to the adaptive-compute literature (accuracy-vs-samples), treating calibration not as a standalone metric but as the candidate mechanism by which training regime affects stopping reliability. Third, by holding the prompt set, sample count, and stopping rule fixed across SFT, IPO, and RLOO, differences in the agreement–calibration–efficiency tradeoff are attributable to the training objective rather than to hyperparameters or evaluation protocol.

4 Results

The value of adaptive stopping turns on two quantities that need not move together: how accurate a policy is at a given compute budget, and how reliably its agreement signal tracks correctness. The first determines what the policy can solve; the second determines whether an agreement-based rule can safely commit early. A policy could be highly accurate yet unsafe to stop early on if its agreement decoupled from correctness, or modestly accurate yet trustworthy to stop on if its agreement were well calibrated. I evaluated all three regimes on both axes, expecting that a reward signal tied to correctness might tighten the link between agreement and correctness. The central finding was a dissociation: RLOO was the least accurate regime yet the best calibrated, so the reliability of agreement-based stopping was not predicted by accuracy. The remainder of this section reports the quantitative comparison along both axes, then examines the per-bin calibration structure to explain why the agreement signal behaved as it did.

4.1 Quantitative Evaluation

Figure 2 plots accuracy against mean samples used, for fixed-k majority voting and for adaptive stopping at $M \in \{2, 3, 4\}$. IPO dominated the accuracy-per-sample frontier, reaching roughly 0.67 accuracy at a mean of six samples under adaptive $M{=}2$. RLOO was the least accurate regime across most of the budget range, while SFT, though weakest at the smallest budgets, rose steeply and became competitive at the largest fixed budget ($k{=}32$). The adaptive operating points sat close to each regime's fixed-k curve but at lower mean sample counts, meaning adaptive stopping matched fixed-k accuracy while drawing fewer samples on average rather than improving accuracy outright; the compute saving therefore carried no accuracy penalty. Because per-prompt accuracy and sample cost depended on the order in which samples arrived, I averaged each adaptive point over $R = 200$ random orderings per prompt, which stabilized these estimates against ordering noise. I did not have multiple training seeds, so cross-seed variance is not reported.

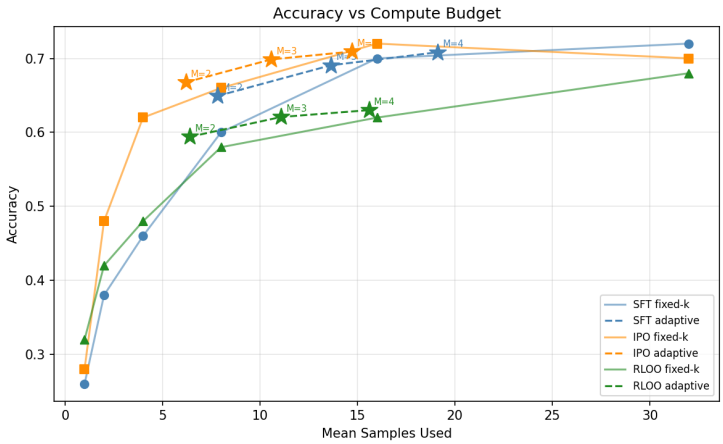

Figure 2: Accuracy vs. mean samples used for fixed-k majority voting (solid) and adaptive stopping

Table 1 reports mean agreement fraction and expected calibration error (ECE) for all conditions. Lowering SFT's sampling temperature sharpened its output distribution, so that agreement rose monotonically from 0.146 at $T = 1.3$ to 0.338 at $T = 0.3$ while ECE fell correspondingly, tracing the calibration frontier reachable by sharpening alone. Both trained regimes fell below this frontier. IPO reached ECE 0.359 at agreement 0.381, and RLOO reached the lowest ECE of any condition, 0.338, narrowly ahead of IPO, at agreement 0.334. At comparable agreement, the trained policies were better calibrated than any temperature setting of SFT achieved, which was the central result of the control plot (Figure 3): training reshaped the agreement–correctness relationship beyond mechanical sharpening, and in the direction of improved rather than degraded calibration, contrary to the concern that RL fine-tuning would make agreement less trustworthy. Placing the two results together isolated the contribution, since RLOO had both the lowest accuracy and the lowest ECE while IPO was the most accurate yet only second-best calibrated. Accuracy and calibration thus ranked the regimes differently, confirming that the safety of agreement-based stopping was a property distinct from how often a policy was correct, and was not recoverable from accuracy alone.

Regime	T	Agreement	ECE
SFT	0.3	0.338	0.382
SFT	0.5	0.332	0.368
SFT	0.7	0.321	0.418
SFT	0.9	0.292	0.434
SFT	1.1	0.238	0.530
SFT	1.3	0.146	0.551
IPO	0.7	0.381	0.359
RLOO	0.7	0.334	**0.359**

Table 1: Mean agreement fraction and expected calibration error

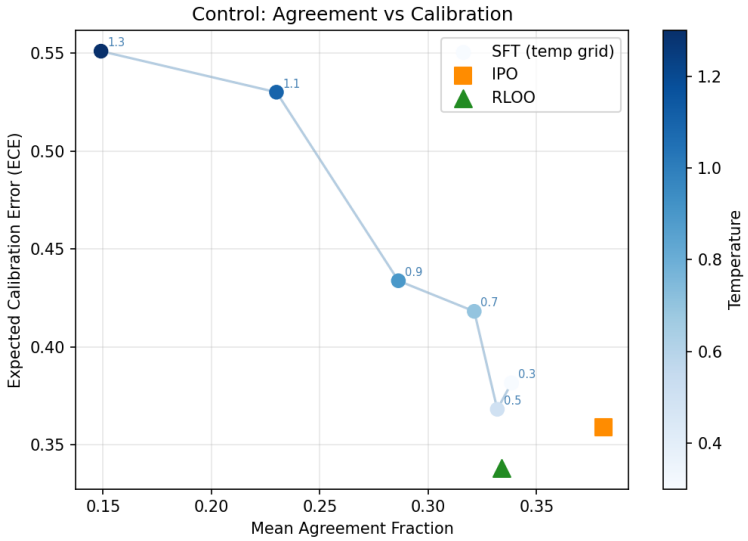

Figure 3: Agreement fraction vs. Calibration (ECE, lower is better

4.2 Qualitative Analysis

The reliability diagrams (Figure 4) reveal the per-bin structure behind the scalar ECE values. For every condition, the accuracy bars sat predominantly above the perfect-calibration diagonal, meaning that within each agreement bin the plurality answer was correct more often than the agreement level implied. All regimes were therefore systematically underconfident on Countdown, and the dominant contribution to ECE was this underconfidence rather than the confident-but-wrong overconfidence the

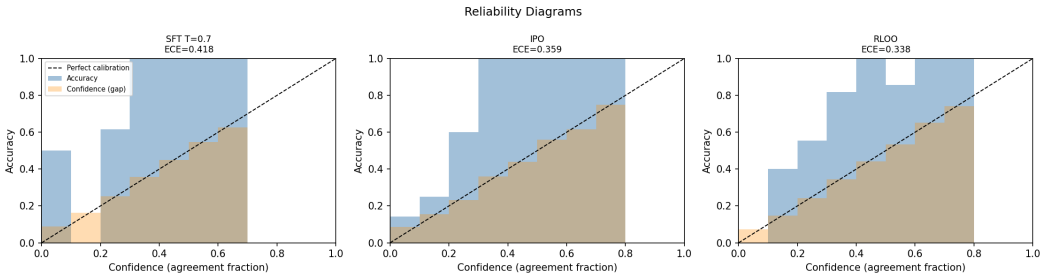

Figure 4: Reliability diagrams for SFT ($T = 0.7$), IPO, and RLOO.

introduction had flagged as the risk for RL policies. That failure mode did not appear: in no condition did high-agreement bins fall substantially below the diagonal, so "best calibrated" for RLOO meant least underconfident rather than that agreement equaled correctness. This pattern was consistent with Countdown's answer structure. Because correct solutions all evaluate to the single target value, correct samples collapsed onto one answer while incorrect samples scattered across many distinct wrong values. The plurality answer was therefore correct more often than the raw agreement fraction suggested, since agreement was diluted by a long tail of one-off wrong answers even when the correct value was modal. This task-specific collapsing of the correct-answer space inflated accuracy relative to agreement and was the most plausible driver of the uniform underconfidence; it also bounded the generality of the finding, since a task whose wrong answers clustered rather than scattered could exhibit the opposite failure mode. A further caveat, visible in every panel, was that the populated bins extended only to roughly 0.7–0.8 on the agreement axis, so very high agreement was rare. The near-unanimous regime that matters most for aggressive early stopping was thus the least sampled, and calibration estimates there carried the most uncertainty. This did not affect the ranking of regimes, which was driven by the well-populated low-to-mid agreement bins, but it tempered any claim about behavior at the extreme of confidence.

5 Discussion

Several limitations bound the scope of these findings. Each non-SFT regime was evaluated at a single sampling temperature (0.7) on a single task. Countdown's verifiable answer structure, in which all correct solutions collapse to one target value, likely drives the systematic underconfidence observed across every regime: correct samples concentrate on a single answer while incorrect ones scatter across many distinct values, so the agreement fraction understates how often the plurality answer is correct. Thus, the calibration results may be specific to tasks with this collapsing structure, and a task whose incorrect answers cluster rather than scatter could exhibit the opposite failure mode. Sweeping temperature for IPO and RLOO and evaluating on tasks with more diffuse answer spaces would test how far these trends generalize. The agreement signal also remains imperfect: because all regimes are underconfident, agreement-based stopping leaves accuracy unexploited, and confidence signals that track correctness more tightly are a natural direction for further work.

In terms of broader impact, the results suggest that agreement-based adaptive stopping can be applied to RL-fine-tuned reasoning policies without the calibration penalty that prior work on reinforcement-learning fine-tuning might lead one to expect, supporting the use of such methods to reduce inference cost in deployed reasoning systems. The same finding carries a caution. Because the reliability of the stopping signal depends on the training regime and on task structure rather than on accuracy alone, a more accurate model is not automatically safer to stop early on, and stopping rules should be validated on the target task rather than transferred across models or domains on the assumption that accuracy implies trustworthy agreement.

The main practical difficulties were training stability and checkpoint management. The RLOO runs were sensitive to configuration and required care to train cleanly, and ensuring a valid controlled comparison meant verifying that all three regimes descended from the same supervised base, since checkpoints initialized from different bases would confound the effect of the training objective with the effect of initialization. The offline design, in which all generations are cached once and replayed

for every metric and stopping simulation, was adopted partly to separate the one-time inference phase from the repeatable analysis phase made the results reproducible and allowed the analysis to be developed and debugged without repeatedly invoking the model on a GPU.

6 Conclusion

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

7 Team Contributions

As a solo project, I, Andrea Nam, performed the end-to-end execution of all components: the implementation of SFT, IPO, and RLOO on the Countdown task, the design and implementation of the adaptive test-time sampling extension, the running and evaluation of all experiments, and the writing of the report.

Changes from Proposal The project followed the proposed scope and methodology. But minor implementation parameters were adjusted during execution such as the sample budget and adaptive thresholds were tuned to the final evaluation.

References

Pranjal Aggarwal, Yiming Yang Aman Madaan, and Mausam. 2023. Let's Sample Step by Step: Adaptive-Consistency for Efficient Reasoning and Coding with LLMs.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Nelson Elhage Sheer El-Showk, Andy Jones, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language Models (Mostly) Know What They Know. arXiv:2207.05221 [cs.CL]

Vint Lee, Pieter Abbeel, and Youngwoon Lee. 2024. DreamSmooth: Improving Model-based Reinforcement Learning via Reward Smoothing. In *The Twelfth International Conference on Learning Representations*.

Yuxiao Qu, Matthew Y. R. Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan, Salakhutdinov, and Aviral Kumar. 2024. Optimizing Test-Time Compute via Meta Reinforcement Fine-Tuning. arXiv:2503.07572 [cs.LG]

Subramanyam Sahoo. 2026. Calibration Collapse Under Sycophancy Fine-Tuning: How Reward Hacking Breaks Uncertainty Quantification in LLMs. arXiv:2604.10585 [cs.LG]

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. 2022. Confident Adaptive Language Modeling. arXiv:2207.07061v2 [cs.CL]

Ksenia Se and Alyona Vert. 2025. *What is test-time compute and how to scale it?* Hugging Face.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. arXiv:2408.03314 [cs.LG]

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv:203.11171v4 [cs.CL]