Hawkeye: Model Collaboration for Efficient Reasoning

Mohamed bin Zayed University of Artificial Intelligence · UCLA · Stanford

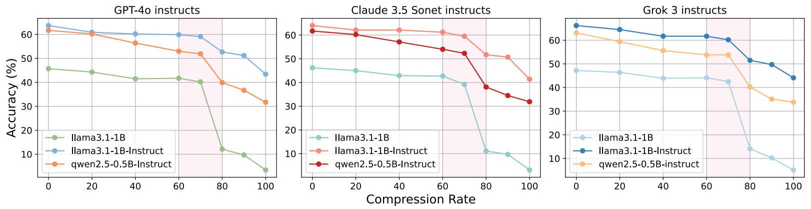
Jianshu She*, Zhuohao Li*, Zhemin Huang, Qi Li, Peiran Xu, Haonan Li, Qirong Ho

Scan for code and models!

COLM

Why reasoning models or your deep research takes so long?

LRMs cosumes lots of <thinking> tokens in CoT reasoning. Why? Majority voting
Is it necessary? No - overthinking commonly happens that exhausting inference
How many tokens are wasted in reasoning? It could be more than you think!



smaller models (LLaMA3.1-1B, Qwen2.5-0.5B) were guided by larger models (GPT-4o, Claude 3.5 Sonnet, Grok 3) to generate CoTs for GSM8K. The CoTs were refined via LLM-assisted feedback by removing (1) repeated, (2) filler-like, and (3) overly fine-grained tokens.

Hawkeye Example

Original Reasoning

Question: *The roots of the equation $2x^2-mx+n=0$ sum to 6 and multiply to 10. What is the value of $m+n$?*

:Okay, so I have this equation here: \(2x^2 - mx + n = 0\). They told me that the sum of the roots is 6 and the product is 10. Hmm, I need to find \(m + n\). Alright, let's see how to approach this. First off, I remember something called Vieta's formulas from algebra class.
To connect the problem statement with the quadratic, I'll use Vieta's relations.
Well, the sum of roots corresponds to \(\tfrac{m}{2}\). I also know the product of roots corresponds to \(\tfrac{n}{2}\). That sets up the system of equations we need to solve.
<Omitted> ...
Oh, so the sum of roots = \(-b/a = -(-m)/2 = m/2\). Product of roots = \(c/a = n/2\) But wait, in the problem statement, they say the sum of the roots is 6 and the product is 10. So that gives us two equations: 1)Sum: \(m/2 = 6\) 2) Product: \(n/2 = 10\) the value is \(\boxed{32}\).

Overthink with more than 600 tokens

Hawkeye Reasoning

(step1) Hawkeye Instruction Generation

: The given quadratic equation has roots that sum to \(\frac{m}{2}\) and multiply to \(\frac{n}{2}\). Given that these sums and products equal \(6\) and \(10\), respectively: \[\frac{m}{2} = 6 \quad \Right arrow \quad m = 12\] \[\frac{n}{2} = 10 \quad \Rightarrow \quad n = 20\] Adding these values together.

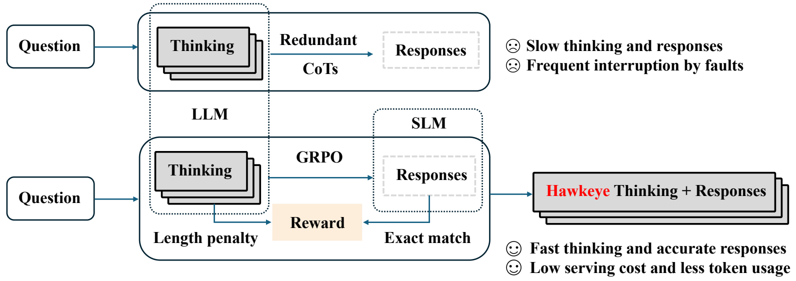
(step2) Hawkeye Response Expansion

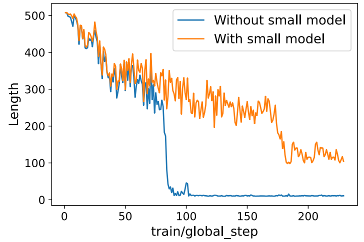
: The given solution correctly identifies the values of \(m\) and \(n\) from the equations derived from the properties of the roots of a quadratic equation. Based on the reasoning provided: 1. The roots of the quadratic equation \(2x^2 - mx + n = 0\) are given by the sum and product of the roots: \[x_1 + x_2 = \frac{-b}{a} = \frac{m}{2} \] \[x_1 \cdot x_2 = \frac{c}{a} = \frac{n}{2} \] . 2. We know from the problem statement that: \[x_1 + x_2 = 6 \implies \frac{m}{2} = 6 \implies m = 12 \] Also, since \(x_1 \cdot x_2 = 10\): \[\frac{n}{2} = 10 \implies n = 20 \] 3. Therefore, we have: \[m + n = 12 + 20 = 32 \] So, the value of \(m + n\) is \(\boxed{32}\).

Efficient reasoning with only 110 output tokens and 72 tokens for CoT generation

An example of HAWKEYE reasoning. The red highlights marks critical tokens that cannot be removed. HAWKEYE generates a high density of critical tokens and reduces token usage by over 70% (600 → 182 tokens).

Methodology - Online Model Collaboration!





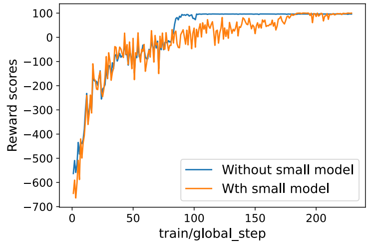
CoT length across training steps. Without a small model, the policy collapses to extremely short CoTs in pursuit of higher rewards

Hawkeye comprises a model collaboration, where the reasoning model produces precise CoT commands while a response model expands instructions and plans for formatting final summarization. Reasoning model are trained based on GRPO

Algorithm 1 GRPO fine-tuning of Model A for *compressed* Chain-of-Thought generation

Require: Policy model π_θ (Model A), frozen response model (Model B), training dataset $\mathcal{D} = \{(q_i, a_i, c_i^{\text{orig}})\}$, instruction prompt s, length penalty weight λ, target compression ratio α (set to 0.3)

while Model A not converged **do**
    for each mini-batch $\{(q_i, a_i, c_i^{\text{orig}})\}_{i=1}^{N} \subset \mathcal{D}$ **do**
        for each (q, a, c^{orig}) in mini-batch **do**
            $c \leftarrow \text{ModelA}(q)$
            $\hat{a} \leftarrow \text{ModelB}(q, c, s)$
            $P \leftarrow \lambda \cdot \max\big(0, \texttt{len}(c) - \alpha \cdot \texttt{len}(c^{\text{orig}})\big)^2$
            $R \leftarrow \mathsf{EM}(\hat{a}, a) - P$
            $\theta \leftarrow \theta + \eta\, \nabla_\theta \log \pi_\theta(c \mid q) \cdot R$
        end for
    end for
end while
return Fine-tuned parameters θ

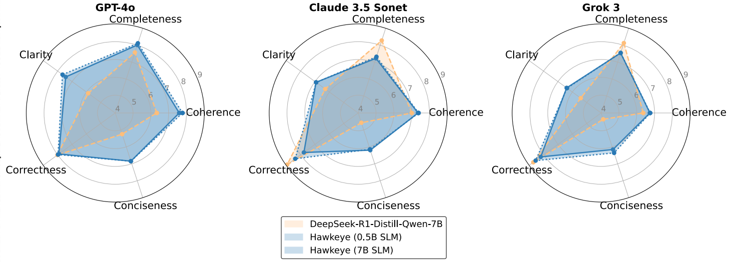


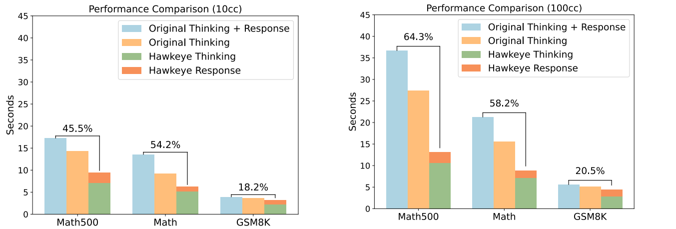
Reward trajectory during training. Both converge to similar reward scores, but a small model can help avoiding reward hacking.

Evaluation results

Dataset	Method	Concurrency	Time per Req (s) ↓	Avg TBT	Tokens ↓
GSM8K	Baseline (Full)	10	3.69	0.013	297.0
	HAWKEYE (CoT only)	10	**2.37**	0.014	**165.7**
	HAWKEYE (Full)	10	**2.81**	0.009	301.7
	Baseline (Full)	100	5.13	0.016	331.0
	HAWKEYE (CoT only)	100	**2.93**	0.017	**171.0**
	HAWKEYE (Full)	100	**3.75**	0.011	332.0
Math500	Baseline (Full)	10	14.95	0.013	1136.54
	HAWKEYE (CoT only)	10	**7.11**	0.015	**471.0**
	HAWKEYE (Full)	10	**8.31**	0.011	**771.4**
	Baseline (Full)	100	36.20	0.021	1463.80
	HAWKEYE (CoT only)	100	**10.74**	0.022	**474.9**
	HAWKEYE (Full)	100	**13.34**	0.017	**765.8**
Math	Baseline (Full)	10	13.09	0.014	942.08
	HAWKEYE (CoT only)	10	**5.14**	0.014	**361.23**
	HAWKEYE (Full)	10	**6.23**	0.011	**565.33**
	Baseline (Full)	100	21.26	0.024	908.08
	HAWKEYE (CoT only)	100	**7.48**	0.022	**332.50**
	HAWKEYE (Full)	100	**9.58**	0.016	**582.10**
AIME	Baseline (Full)	10	93.48	0.016	5943.90
	HAWKEYE (CoT only)	10	**66.60**	0.016	**4168.20**
	HAWKEYE (Full)	10	**68.90**	0.015	**4722.50**

Benchmark Performance: Time per Req denotes average latency per request and Tokens denotes the token counts





Dataset	**Method**	**Acc**	**Tokens**	**Compression Rat**
GSM8K	CoT-Valve (QwQ-32B-Preview)	95.1% → 94.0%	741.1 → 352.8	52.4
	Hawkeye (DeepSeek-R1-Distill-Qwen-7B)	90.7% → 88.9%	331.0 → 171.0	48.3
MATH	O1-Pruner (QwQ-32B-Preview)	90.6% → 91.0%	2191.0 → 1385.0	36.7
	Hawkeye (DeepSeek-R1-Distill-Qwen-7B)	92.3% → 89.4%	942.0 → 361.0	61.7
MATH500	Hawkeye (DeepSeek-R1-Distill-Qwen-7B)	80.1% → 75.5%	1463.0 → 474.9	67.5

Hawkeye v.s. Baseline: 7B models can still perform similar with 32B models