

Dubito, Ergo Sum: Self-Evolving Adaptive Reasoning via Intrinsic-Weighted Group Optimization

Anonymous ACL Submission

Abstract

Large Reasoning Models (LRMs) excel at complex tasks using Chain-of-Thought prompting but suffer from 'overthinking,' often inefficiently allocating costly reasoning resources to simple queries. Existing adaptive methods typically rely on opaque reinforcement learning strategies based on reasoning-length penalties, which lack both interpretability and intrinsic grounding. We argue that true efficiency stems from a model's internalized self-awareness of task complexity. In this work, we introduce **Dubito-Pro**, a framework where the LLM autonomously selects between "Fast" and "Slow" thinking modes based on the input context, without external classifiers or inference-time intervention. Our core insight is that Entropy Variance serves as a high-fidelity supervision signal for cognitive struggle during training. To instill this capability, we propose Intrinsic-Weighted Group Relative Policy Optimization (I-GRPO). Unlike standard RL approaches that reward solely outcome correctness, I-GRPO introduces a Cognitive Alignment Reward, computed post-hoc during training. This mechanism penalizes the model for selecting a 'Fast' path on high-variance (ambiguous) queries, effectively teaching it to anticipate its own uncertainty. Extensive experiments on a mixed-difficulty benchmark demonstrate that Dubito-Pro acquires a robust, intuitive switching policy. Reducing token costs by 80% on simple tasks and improving overall accuracy by 7.47%, this method establishes a new Pareto frontier for efficiency and accuracy.

1 Introduction

The paradigm of Chain-of-Thought (CoT) prompting has revolutionized the capabilities of Large Language Models (LLMs), enabling them to solve complex mathematical and logical problems by generating detail intermediate reasoning steps (Achiam et al., 2023; Anil et al., 2023; Dubey et al., 2024). Recent scaling of this paradigm—exemplified by

models like OpenAI's o1 and DeepSeek-R1—has pushed performance to new heights. However, this progress comes with a significant cost: "**overthinking**." Current LRMs tend to apply exhaustive, token-heavy reasoning processes indiscriminately, even for trivial queries. This not only incurs prohibitive inference latency and computational costs but can also degrade performance on simple tasks by inducing hallucinations or unnecessary complications (Jaech et al., 2024; Guo et al., 2024).

To address this efficiency bottleneck, the field has turned toward Adaptive Reasoning, aiming to dynamically allocate effort based on problem difficulty. Early approaches relied on static heuristics or external classifiers, which often lack the granularity to handle diverse queries (Shinn et al., 2023; Ong et al., 2024). More recent state-of-the-art methods, such as O1-Pruner (Luo et al., 2025a) and Adapt-Think (Jaech et al., 2024), utilize Reinforcement Learning (RL) to train models to switch between reasoning formats. While effective, these methods fundamentally operate as "black boxes." They optimize for extrinsic outcome rewards without modeling the underlying cognitive process.

In this work, we advocate for a paradigm shift from reward-driven adaptation to mechanism-driven self-awareness. We posit that a truly intelligent agent should not merely mimic efficient behaviors to maximize a score; it should possess an intrinsic awareness of its own cognitive stability.

To this end, we introduce **Dubito-Pro**, an interpretable framework that enables LLMs to dynamically switch between "Fast Thinking" and "Slow Thinking" modes based on intrinsic signals. Our core insight is that token-level entropy variance serves as a robust "thermometer" for cognitive instability. Unlike confidence scores (Fu et al., 2025), which can be noisy, sharp fluctuations in entropy variance reliably signal two critical states: logical confusion (where the model vacillates between reasoning paths) and degenerative repetition (where

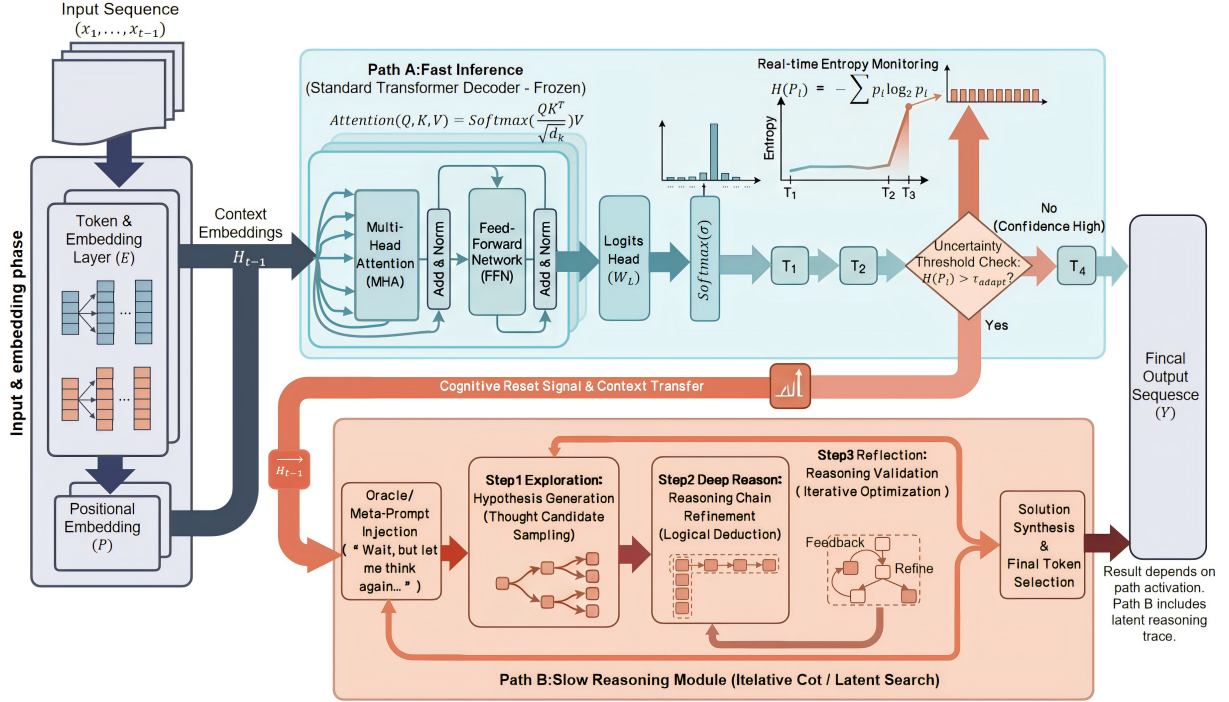


Figure 1: Existing approaches face a dilemma: they either always **overthink** or always **underthink**. Our goal is to achieve intelligent **self-adaptation**.

the model becomes trapped in a repetitive loop).

To effectively internalize this capability, we propose **Intrinsic-Weighted Group Relative Policy Optimization (I-GRPO)**, a novel online RL algorithm. Standard GRPO optimizes solely for correctness. In contrast, I-GRPO introduces a Cognitive Alignment Reward, which incentivizes the model to align its behavioral strategy with its internal state—rewarding "Slow Thinking" when confused and "Fast Thinking" when confident. Additionally, I-GRPO applies Uncertainty-Weighted Advantage estimation, dynamically amplifying the learning signal at critical decision points where cognitive struggle is highest. This allows the model to self-evolve a robust switching policy through exploration, rather than merely imitating static datasets.

Our contributions are summarized as follows:

- 1) **Interpretability via Intrinsic Mechanism:** We propose Dubito-Pro, the first adaptive framework driven by the transparent mechanism of entropy variance.
- 2) **Intrinsic-Weighted GRPO (I-GRPO):** We introduce I-GRPO, an online RL algorithm that integrates the model's intrinsic cognitive state into the reward function. By promoting "cognitive alignment," I-GRPO achieves more robust policy learning than standard outcome-based methods.
- 3) **State-of-the-Art Efficiency:** Extensive ex-

periments on a mixed-difficulty benchmark demonstrate that Dubito-Pro establishes a new Pareto frontier. It exceeds the accuracy of reasoning-heavy models on complex tasks while reducing token usage by nearly 80% on simpler, significantly outperforming baselines like AdaptThink and O1-Pruner.

2 Related Work

2.1 Efficient Reasoning in LLMs

The Chain-of-Thought (CoT) paradigm (Wei et al., 2022) and its extensions, such as Self-Consistency (Wang et al., 2022) and Tree of Thoughts (Yao et al., 2023), have significantly enhanced LLM performance on complex tasks. However, these methods incur high computational costs by applying exhaustive reasoning indiscriminately. While general efficiency techniques like speculative decoding (Leviathan et al., 2023) or model pruning (Cheng et al., 2024) optimize inference speed, they do not address the strategic allocation of cognitive effort. Our work targets this strategic efficiency, aiming to reduce token consumption at the semantic planning level rather than the system execution level.

2.2 Adaptive Reasoning Frameworks

Recent efforts have sought to dynamically adjust reasoning complexity. Early approaches relied on static heuristics or external classifiers to route

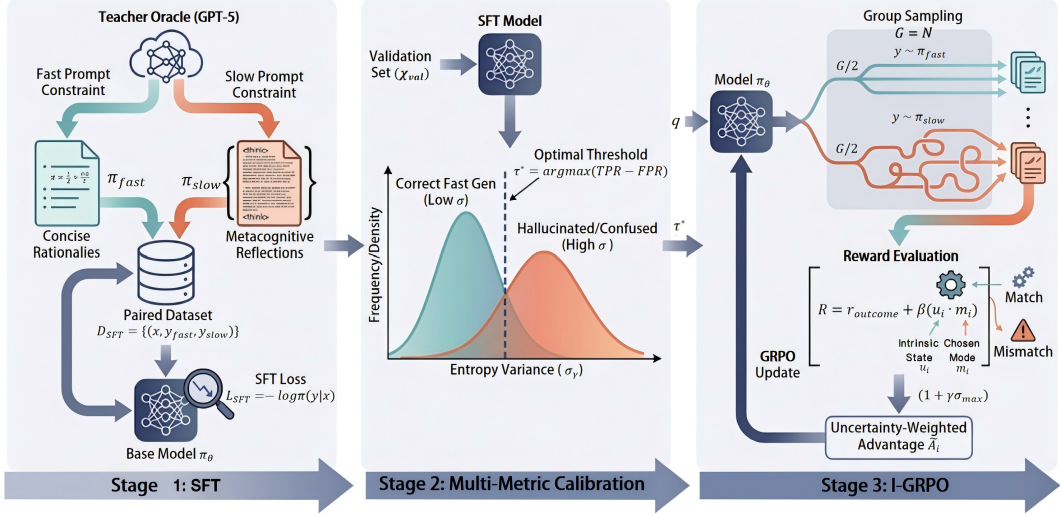


Figure 2: An Overview of the Dubito Method: Uncertainty-Driven Adaptive Thinking.

queries to different prompts or models (Xu et al., 2025; Ong et al., 2024), often lacking granularity. More advanced methods, such as O1-Pruner (Luo et al., 2025a) and AdaptThink (Jaech et al., 2024), employ RL to train models to select reasoning formats. While effective, these methods fundamentally operate as "black boxes," optimizing for extrinsic outcome rewards without modeling the underlying cognitive process. Consequently, their decision-making remains opaque. Other works like DeepConf (Fu et al., 2025) and ConCISE (Qiao et al., 2025) utilize confidence scores for inference-time early stopping. While interpretability is improved, these are external interventions rather than internalized model capabilities. Dubito-Pro bridges this gap by grounding adaptive behavior in an intrinsic, interpretable mechanism (entropy variance) and internalizing it through training, ensuring both transparency and robustness.

2.3 Uncertainty and Policy Optimization

Uncertainty estimation via token-level probability (entropy) or semantic consistency has been widely used for hallucination detection (Wastl et al., 2025; Vazhentsev et al., 2025; Uesato et al., 2022). We extend this by identifying entropy variance as a novel indicator of cognitive struggle and repetition loops. In terms of training, while Direct Preference Optimization (DPO) (Rafailov et al., 2023) and Simple Preference Optimization (SimPO) (Meng et al., 2024) have streamlined offline alignment, they lack the exploration capabilities required for self-evolving strategies. GRPO (Shao et al., 2024) introduces online group-based optimization but fo-

cuses solely on outcome correctness. Our proposed I-GRPO innovates by integrating intrinsic cognitive states into the reward function, enabling the model to learn a policy that aligns its reasoning depth with its internal uncertainty, a paradigm we term Cognitive Alignment.

3 Methodology

We propose Dubito-Pro, a framework designed to internalize adaptive reasoning capabilities into LLMs. Our approach is structured around three key components: a Dual-Mode Reasoning mechanism driven by SFT, a variance-based uncertainty quantification metric, and a three-stage training pipeline that culminates in Intrinsic-Weighted Group Relative Policy Optimization (I-GRPO). The overall process is illustrated in Figure 2.

3.1 Dual-Mode Reasoning Framework

We define the reasoning space \mathcal{Y} as a composition of two distinct policies derived from the same base model parameters θ , triggered via specific control tokens. The Fast Thinking (π_{fast}) policy generates a concise rationale followed immediately by the answer, optimizing for efficiency on simple tasks. Conversely, the Slow Thinking (π_{slow}) policy engages in detailed decomposition, self-verification, and error correction, prioritizing accuracy for complex queries.

3.2 Intrinsic Signal: Entropy Variance

Unlike previous works that rely on raw confidence scores, we posit that cognitive struggle—and degenerative repetition—is best characterized by the

instability of the model’s predictive distribution. At any decoding step t , given context $x_{<t}$, we first calculate the normalized entropy $H_n(p_n)$ of the probability distribution over the vocabulary V :

$$H_n(p_n) = \frac{-\sum_{v \in V} p_t(v) \log_t(v)}{\log |V|}. \quad (1)$$

To capture the dynamics of uncertainty, we compute the Entropy Standard Deviation (σ) within a local sliding window of size w (e.g., $w=10$):

$$\sigma_t(H) = \sqrt{\frac{1}{w} \sum_{j=0}^{w-1} (H_n(p_{t-j}) - \bar{H}_t)^2}, \quad (2)$$

where \bar{H}_t is the mean entropy within the window. This metric serves as the governing signal for our switching logic. During Fast Thinking generation, $\sigma_t(H)$ exceeds a calibrated threshold τ , the process is halted and restarted using Slow Thinking:

$$\text{Policy}(x) = \begin{cases} \pi_{\text{slow}}(x), & \text{if } \exists t, \sigma_t(H) > \tau \\ \pi_{\text{fast}}(x), & \text{otherwise} \end{cases}. \quad (3)$$

Theoretically, while stable low entropy indicates confidence and stable high entropy indicates ignorance (Huang et al., 2025; Agarwal et al., 2025), a high variance (σ_t) uniquely signals oscillation between conflicting reasoning paths or the onset of degenerative repetition loops, where the model vacillates at the boundaries of repetitive phrases. Triggering a switch at these moments acts as a "Cognitive Reset," effectively breaking the loop.

3.3 Stage 1: Stylized Supervised Fine-Tuning (S-SFT)

To initialize the dual capabilities, we fine-tune the base model on a paired dataset $\mathcal{D}_{\text{SFT}} = \{(x, y_{\text{fast}}, y_{\text{slow}})\}$. These pairs are synthesized from a strong teacher model using style-specific prompts. The objective is the standard negative log-likelihood loss:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(x,y) \sim \mathcal{D}_{\text{SFT}}} [\log \pi_{\theta}(y | x)]. \quad (4)$$

We observe that, following the first stage of cold-start training, the model has preliminarily acquired the ability to automatically switch reasoning paths.

3.4 Stage 2: Multi-Metric Calibration (M-Calibrate)

We determine the optimal threshold τ by running the SFT model on a validation set. We collect the

trajectory of $\sigma_t(H)$ for both correct and incorrect generations and select τ to maximize the separation between successful Fast Thinking and necessary Slow Thinking using Youden’s J statistic:

$$\tau^* = \arg \max_{\tau} (\text{TPR}(\tau) - \text{FPR}(\tau)). \quad (5)$$

Here, True Positives (Youden, 1950) are defined as instances where the model is incorrect (or repetitive) and $\sigma_t(H) > \tau$.

Algorithm 1 Intrinsic-Weighted GRPO (I-GRPO)

Require: Dataset \mathcal{D} , Policy π_{θ} , Reference Policy π_{ref} , Threshold τ
Require: Hyperparameters: Group size G , β (align weight), γ (advantage scale), ϵ (clip)

- 1: **while** not converged **do**
- 2: Sample batch of queries $Q \sim \mathcal{D}$
- 3: **for** each query $q \in Q$ **do**
- 4: **Step 1: Hybrid Group Sampling**
- 5: Sample $G/2$ trajectories via $\pi_{\theta}(\cdot | q, \text{Fast})$
- 6: Sample $G/2$ trajectories via $\pi_{\theta}(\cdot | q, \text{Slow})$
- 7: Combine to form group $\mathcal{O} = \{o_1, \dots, o_G\}$
- 8: **Step 2: Cognitive Alignment Reward Evaluation**
- 9: **for** each trajectory $o_i \in \mathcal{O}$ **do**
- 10: Calculate correctness: $r_{\text{out}} \leftarrow \mathbb{I}(o_i \text{ is correct})$
- 11: Calculate efficiency: $r_{\text{eff}} \leftarrow 1/\text{Length}(o_i)$
- 12: Compute peak entropy: $\sigma_{\text{max}}^{(i)} \leftarrow \max_t \sigma_t(H)^{(i)}$
- 13: Determine Intrinsic State u_i and Mode m_i :
- 14: $u_i \leftarrow 1$ (Confused) if $\sigma_{\text{max}}^{(i)} > \tau$ else -1 (Confident)
- 15: $m_i \leftarrow 1$ (Slow) if $o_i \sim \pi_{\text{slow}}$ else -1 (Fast)
- 16: Alignment Reward: $r_{\text{align}} \leftarrow u_i \cdot m_i$
- 17: Total Reward: $R(o_i) \leftarrow r_{\text{out}} + \lambda_1 r_{\text{eff}} + \beta r_{\text{align}}$
- 18: **end for**
- 19: **Step 3: Uncertainty-Weighted Advantage Estimation**
- 20: Compute group statistics: mean \bar{R} and std S of $\{R(o_i)\}$
- 21: **for** each trajectory $o_i \in \mathcal{O}$ **do**
- 22: Standard Advantage: $A_i \leftarrow (R(o_i) - \bar{R}) / (S + \epsilon)$
- 23: Weighted Advantage: $\hat{A}_i \leftarrow (1 + \gamma \cdot \sigma_{\text{max}}^{(i)}) \cdot A_i$
- 24: **end for**
- 25: **Step 4: Policy Update**
- 26: Compute probability ratio: $\rho_i = \pi_{\theta}(o_i | q) / \pi_{\theta_{\text{old}}}(o_i | q)$
- 27: Optimize objective:
- 28:
$$\mathcal{L} \leftarrow -\frac{1}{G} \sum_{i=1}^G \left[\min(\rho_i \hat{A}_i, \text{clip}(\rho_i, 1 - \epsilon, 1 + \epsilon) \hat{A}_i) \right. \\ \left. - \beta_{\text{KL}} D_{\text{KL}}(\pi_{\theta} || \pi_{\text{ref}}) \right]$$
- 29: **end for**
- 30: Update π_{θ} via gradient descent
- 31: **end while**

3.5 Stage 3: Intrinsic-Weighted GRPO (I-GRPO)

Standard RL methods maximize extrinsic rewards, often ignoring the underlying cognitive process. We introduce I-GRPO to align the model’s behavior with its intrinsic cognitive state.

For each input query q , we employ a Hybrid Group Sampling strategy, sampling a group of G outputs $\{o_1, \dots, o_G\}$ where $G/2$ trajectories are sampled using π_{fast} and $G/2$ using π_{slow} . This ensures comparative learning across modes within the same optimization step.

To incentivize the model to respect its uncertainty, we introduce a Cognitive Alignment Reward $R_{\text{align}} = \beta \cdot (u_i \cdot m_i)$. Let $\sigma_{\text{max}}^{(i)}$ be the peak entropy

variance of trajectory o_i . We define the uncertainty state $u_i \in \{1, -1\}$ (where 1 represents confusion, i.e., $\sigma_{\max}^{(i)} > \tau$) and mode $m_i \in \{1, -1\}$ (where 1 represents Slow Thinking). The total reward is defined as:

$$R(o_i) = \mathbb{I}(o_i \text{ is correct}) + \beta \cdot (u_i \cdot m_i) - \lambda |o_i|. \quad (6)$$

The term $u_i \cdot m_i$ rewards the model for "thinking slow" when confused and "thinking fast" when confident, while penalizing misalignment.

Finally, to focus learning on critical decision points where the model is on the "cognitive boundary," we apply Uncertainty-Weighted Advantage Estimation. The standard advantage A_i is scaled by the magnitude of the uncertainty variance:

$$\hat{A}_i = \left(1 + \gamma \cdot \sigma_{\max}^{(i)}\right) \cdot \frac{R(o_i) - \bar{R}}{\text{std}(R)}, \quad (7)$$

where γ is a scaling hyperparameter. This ensures that gradients are dominated by samples where the model experienced significant cognitive struggle. The final optimization objective follows the standard GRPO formulation using these weighted advantages:

$$\begin{aligned} \mathcal{L}_{\text{I-GRPO}}(\theta) = \mathbb{E}_{q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(O|q)} & \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \right. \\ & \sum_{t=1}^{|o_i|} \left(\min \left\{ \frac{\pi_{\theta}(o_{i,t} | q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} | q, o_{i,<t})} \hat{A}_{i,t}, \right. \right. \\ & \left. \left. \text{clip} \left(\frac{\pi_{\theta}(o_{i,t} | q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} | q, o_{i,<t})}, 1-\varepsilon, 1+\varepsilon \right) \right. \right. \\ & \left. \left. \hat{A}_{i,t} \right\} - \beta D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{ref}}) \right]. \quad (8) \end{aligned}$$

The I-GRPO training procedure is outlined in Algorithm 1.

4 Experiments

4.1 Experimental Setup

To rigorously evaluate the efficiency-accuracy trade-off and the interpretability of our framework, we conduct experiments across a diverse set of reasoning tasks ranging from simple factual recall to complex competition-level mathematics.

Base Model and Training Data. We utilize Qwen3-8B (Yang et al., 2025) and DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025) as our primary

base model for all experiments to ensure a fair comparison. For the Stage 1 (S-SFT) and Stage 3 (I-GRPO) training phases, we construct a high-quality paired dataset derived from a mixture of GSM8K (Cobbe et al., 2021), DeepScaleR (Luo et al., 2025b), and CommonsenseQA (Talmor et al., 2019). We employ GPT-5 as a teacher model to synthesize pairs of Fast Thinking (concise, direct rationale) and Slow Thinking (detailed, metacognitive reflection) responses for approximately 5k training queries, ensuring distinct stylistic boundaries between the two modes.

Mixed-Difficulty Benchmarks. We curate a Mixed-Difficulty Benchmark comprising three distinct datasets to evaluate performance across the full spectrum of cognitive complexity. To assess performance on simple, "System 1" tasks, we utilize GSM8K, where the primary objective is to prioritize efficiency via Fast Thinking with near-zero mode switching. For intermediate tasks requiring multi-step implicit reasoning, we employ MATH500 (Lightman et al., 2023); these serve as the primary testing ground for the model's ability to dynamically switch modes based on specific logical hurdles. At the highest level of difficulty, we include AIME24 (30 Olympiad-level math problems), representing complex "System 2" tasks where the model should predominantly utilize Slow Thinking to maximize accuracy. Finally, to test the generalization of our intrinsic uncertainty mechanism to unseen task types, we include StrategyQA (Geva et al., 2021) and MMLU (Hendrycks et al., 2020) as an Out-of-Distribution (OOD) benchmark.

Baselines. We compare Dubito-Pro against three categories of baselines, all utilizing the DeepSeek-R1-Distill-Qwen-7B base model. First, we establish static performance bounds using Fast-Only and Slow-Only models, which represent the lower bound for cost and the upper bound for accuracy, respectively. Second, we benchmark against state-of-the-art adaptive methods: (1) O1-Pruner (Luo et al., 2025a), employs offline PPO-style length-harmonizing fine-tuning with a joint length-accuracy reward to reduce inference overhead. (2) LASER (Liu et al., 2025), a RL-based length-step reward method, rewards correct responses within a target length to balance LRMs' performance and token efficiency. (3) AdaptThink (Zhang et al., 2025a), using RL algorithm to teach models to adaptively choose between thinking and

Method	GSM8K			MATH500			AIME2024			Average		
	Acc \uparrow	Length \downarrow	AES \uparrow	Acc \uparrow	Length \downarrow	AES \uparrow	Acc \uparrow	Length \downarrow	AES \uparrow	Acc \uparrow	Length \downarrow	AES \uparrow
<i>Base Model</i>												
<i>DeepSeek-R1-Distill-Qwen-7B</i>	90.22	996	0	87.54	2877	0	43.33	9645	0	73.69	4456	0
<i>Static Baselines</i>												
<i>FastThink (Fast-SFT Only)</i>	89.23	143	<u>0.80</u>	84.54	907	<u>0.51</u>	30.00	7262	-1.29	67.92	2771	0.01
<i>SlowThink (Slow-SFT Only)</i>	93.48	1212	-0.11	<u>92.22</u>	3710	-0.13	53.33	11879	0.46	79.67	5600	0.07
<i>Dubito-SFT (SFT Only)</i>	89.99	238	0.75	89.28	2155	0.31	46.67	11046	0.09	75.21	4479	0.38
<i>Adaptive Baselines</i>												
<i>Laser-D (Liu et al., 2025)</i>	<u>92.49</u>	1032	0.04	92.36	1983	0.48	<u>56.67</u>	11386	0.74	<u>80.50</u>	4800	0.41
<i>Laser-DE (Liu et al., 2025)</i>	92.72	991	0.09	91.70	1908	0.48	53.33	<u>8334</u>	0.83	79.25	3744	0.46
<i>O1-Pruner (Luo et al., 2025a)</i>	87.60	428	0.43	86.80	2534	0.08	49.2	9719	0.40	74.53	4227	0.30
<i>AdaptThink (Zhang et al., 2025a)</i>	91.00	309	0.72	92.00	1875	0.50	55.60	8699	<u>0.95</u>	79.53	<u>3627</u>	<u>0.72</u>
<i>Ours</i>												
<i>Dubito-Pro-7B (DPO)</i>	90.45	<u>203</u>	<u>0.80</u>	90.04	2254	0.30	53.33	9548	0.70	77.94	4001	0.60
<i>Dubito-Pro-7B (I-GRPO)</i>	92.07	203	0.86	91.40	<u>1774</u>	0.52	60.00	9495	1.17	81.16	3824	0.85

Table 1: **Performance comparison of the DeepSeek-R1-Distill-Qwen-7B base model across task difficulty levels.** We report Pass@1 Accuracy (%) and Average Token Length. The AES metric quantifies the "return on compute," where a higher value indicates superior efficiency in extracting cognitive value. **Bold** indicates the best performance; Underlined indicates the second best.

non-thinking modes based on problem difficulty. Finally, to isolate the contributions of our specific algorithms, we evaluate two ablation variants: Dubito-SFT, using only threshold-based switching without RL; and Dubito-DPO, which using standard offline preference optimization.

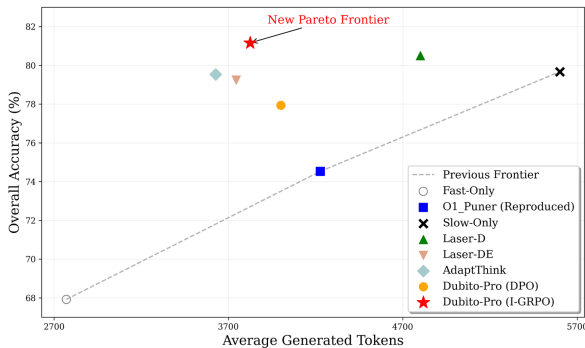


Figure 3: The Efficiency-Accuracy Pareto Frontier on the Mixed-Difficulty Benchmark.

Evaluation Metrics. We adopt average accuracy, average response length, and the Accuracy-Efficiency Score (AES) (Luo et al., 2025a) as core metrics to evaluate whether the model strikes an optimal balance between reasoning accuracy and inference cost (reflected by response length).

4.2 RQ1: Pareto Frontier Analysis

To answer the first research question regarding the efficiency-accuracy trade-off, we analyze the performance of Dubito-Pro against all baselines across the Mixed-Difficulty Benchmark. The efficiency-

accuracy Pareto frontier results are visualized in Figure 3 and detailed numerically in Table 1.

Overall Pareto Dominance. As illustrated in Figure 3, Dubito-Pro establishes a new, superior Pareto frontier compared to existing methods. In the scatter plot where the x-axis represents the Average Token Cost and the y-axis represents Accuracy, Dubito-Pro is positioned nearly in the upper-left of all competing baselines. This indicates that, Dubito-Pro achieves higher accuracy while consuming fewer tokens. Notably, Dubito-Pro exceeds the peak performance of the Slow-Only baseline on complex reasoning tasks while reducing the overall token consumption by approximately 32%.

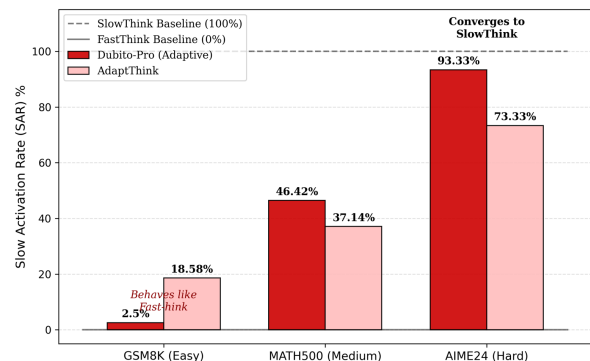


Figure 4: The Slow Activation Rate (SAR) shifting dramatically based on task difficulty.

Performance on Stratified Difficulty. A deeper inspection of Figure 4 reveals how Dubito-Pro achieves this efficiency. On Easy tasks (GSM8K),

Method	GSM8K			MATH500			AIME2024			Average		
	Acc \uparrow	Length \downarrow	AES \uparrow	Acc \uparrow	Length \downarrow	AES \uparrow	Acc \uparrow	Length \downarrow	AES \uparrow	Acc \uparrow	Length \downarrow	AES \uparrow
<i>Base Model</i>												
<i>Qwen3-8B</i>	95.00	2354	0	92.86	5116	0	76.67	16434	0	<u>88.17</u>	7968	0
<i>Static Baselines</i>												
<i>FastThink (Fast-SFT Only)</i>	92.19	130	<u>0.79</u>	84.84	911	0.39	40.00	7538	-1.85	72.34	2859	-0.22
<i>SlowThink (Slow-SFT Only)</i>	95.38	2793	-0.17	93.60	5461	-0.04	66.67	18350	-0.76	85.21	8868	-0.32
<i>Dubito-SFT (SFT Only)</i>	94.69	666	0.70	92.72	2733	0.46	<u>73.33</u>	12303	<u>0.03</u>	86.91	5234	<u>0.39</u>
<i>Ours</i>												
<i>Dubito-Pro-8B (DPO)</i>	<u>95.07</u>	690	0.71	<u>93.14</u>	2318	<u>0.55</u>	70.00	13741	-0.27	86.07	5583	0.33
<i>Dubito-Pro-8B (I-GRPO)</i>	95.38	<u>482</u>	0.81	92.72	<u>2176</u>	0.57	76.67	<u>9670</u>	0.41	88.26	<u>4109</u>	0.60

Table 2: Performance comparison of the Qwen3-8B base model across task difficulty levels.

Component	Variant / Method	Mechanism	Acc \uparrow	Length \downarrow	Wasted Switch Rate \downarrow
Full Model	Dubito-Pro-7B	I-GRPO + Variance	81.16	3874	12.16
Algorithm Ablation	<i>w/o</i> Intrinsic Reward	Standard GRPO	81.64	5261	46.23
	<i>w/o</i> Online RL	DPO	77.94	4001	27.36
	<i>w/o</i> RL	SFT Only	75.21	4479	34.96
Metric Ablation	<i>w/o</i> Variance	Max Entropy (H_{max})	78.25	4346	24.12
	<i>w/o</i> Variance	Entropy Diff (ΔH)	79.63	4263	20.03

Table 3: Ablation study on training algorithms and uncertainty metrics. We evaluate variants of Dubito-Pro on the Mixed-Difficulty Benchmark. "Std-GRPO" removes the Cognitive Alignment Reward and Weighted Advantage. "DPO" replaces online RL with offline preference learning. "Max Entropy" and "Entropy Diff" replace our variance-based trigger with maximum entropy and entropy difference, respectively.

Dubito-Pro behaves almost identically to the FastThink model (0%). It achieves an extremely low Slow Activation Rate (SAR < 3%), demonstrating that the entropy variance metric successfully identifies confident, stable generation and inhibits unnecessary CoT expansion. This contrasts sharply with the SlowThink baseline, which expends thousands of tokens on these trivial queries without yielding significant accuracy gains. Conversely, on Hard tasks (AIME24), Dubito-Pro effectively converges to the performance profile of the SlowThink model (100%). The high SAR (> 93%) in these subsets confirms that the model correctly perceives its own cognitive instability during complex problem-solving and proactively switches to the rigorous SlowThink mode. Crucially, comparison with AdaptThink reveals significant behavioral differences. AdaptThink’s activation curve is notably flatter, maintaining an SAR of $\sim 18\%$ on easy tasks and only $\sim 73\%$ on hard tasks. This suggests that the constrained optimization objective used by AdaptThink is a less precise proxy for cognitive struggle than our entropy variance.

Comparison with Adaptive Baselines. Comparison with Adaptive Baselines. Dubito-Pro demonstrates significant advantages over state-of-the-art

adaptive baselines. As shown in the table, Dubito-Pro (I-GRPO) achieves the highest average accuracy of 81.16% and an AES of 0.85, outperforming both Laser and O1-Pruner. While prior methods like Laser rely on length-based step rewards or problem-level classification, our results suggest that token-level intrinsic monitoring offers superior sensitivity to reasoning quality. More importantly, Dubito-Pro consistently outperforms AdaptThink. Although AdaptThink implements adaptive switching, its reliance on outcome-based rewards leads to suboptimal behaviors on complex tasks. Specifically, we observe that AdaptThink tends to "under-think" on challenging problems (e.g., in AIME2024, where it trails Dubito-Pro by 4.4%). Furthermore, the I-GRPO algorithm enables a more decisive policy, reducing the rate of "wasted switches" by 5.3%. This confirms that aligning the policy with intrinsic cognitive states yields more robust and resource-efficient adaptation than optimizing for extrinsic correctness alone.

4.3 RQ2: Ablation Studies

To isolate the contributions of our algorithmic innovations and metric selection, we conduct a series of ablation studies. These experiments aim to deter-

mine whether the performance gains of Dubito-Pro stem from the I-GRPO training paradigm or the Entropy Variance signal itself. The detail results are summarized in Table 3.

Method	StrategyQA			MMLU		
	Acc	Length	AES	Acc	Length	AES
<i>DeepSeek-RI-Distill-Qwen-7B</i>	59.64	895	0	64.52	1486	0
<i>FastThink</i>	54.17	113	0.84	55.83	168	0.21
<i>SlowThink</i>	59.61	1302	-0.45	65.03	2038	-0.35
<i>Dubito-SFT-7B</i>	59.91	275	0.71	63.78	317	0.73
<i>Dubito-Pro-7B (DPO)</i>	59.91	156	0.84	64.73	241	0.85
<i>Dubito-Pro-7B (GRPO)</i>	61.35	155	0.91	65.08	201	0.89

Table 4: The performance of *Dubito-Pro* on the out-of-distribute test set StrategyQA and MMLU.

Impact of Training Algorithms (I-GRPO vs. Baselines): We evaluate the full Dubito-Pro model against two variants: one trained via Standard GRPO (optimizing solely for outcome correctness and length) and one via DPO (offline preference optimization). While the Standard GRPO baseline matches Dubito-Pro’s accuracy on complex tasks, its efficiency is significantly compromised. Lacking the Cognitive Alignment Reward, it falls prey to "reward hacking," defaulting to Slow Thinking to ensure correctness. Conversely, the DPO variant exhibits reduced robustness. Although it acquires general switching behaviors from SFT data, it struggles with edge cases in medium-difficulty scenarios. We attribute this to the distribution shift inherent in offline training, where the model’s internal uncertainty patterns during inference diverge from the static SFT reference.

Impact of Uncertainty Metrics (Variance vs. Magnitude): To validate our choice of Entropy Variance (σ), we replace it with two common alternatives as the switching trigger: Max Entropy (used in general uncertainty estimation (Xia et al., 2025)) and Entropy Difference (used in ADR (Zhang et al., 2025b)). We evaluate these on the Mixed-Difficulty Benchmark to measure the Wasted Switches Rate. The Max Entropy baseline performs poorly, yielding a high WSR of 35%; qualitative analysis reveals that it frequently spikes during benign token selection (e.g., choosing between synonyms like "Therefore" and "Thus"), thereby confusing lexical diversity with genuine cognitive struggle. Similarly, while Entropy Difference effectively detects sudden reasoning shifts, it fails to capture oscillatory confusion, where the model vacillates between plausible logical paths without exhibiting a sharp jump in entropy. In contrast, our Entropy Vari-

ance metric achieves the lowest WSR of 12%. By quantifying distributional stability over a sliding window, it effectively filters out momentary lexical noise while amplifying signals of persistent cognitive hesitation or degenerative loops, demonstrating that temporal instability—rather than raw uncertainty magnitude—serves as the superior signal for adaptive control.

4.4 RQ3: Generalization to OOD Tasks

A fundamental challenge in adaptive reasoning is Out-of-Distribution (OOD) generalization. Deep learning-based methods often risk overfitting to the surface-level features of their training data. Consequently, these models frequently falter when faced with tasks that diverge significantly from the training distribution. To rigorously evaluate whether Dubito-Pro’s intrinsic mechanism transfers effectively to unseen domains, we test the models on two benchmarks that were strictly excluded from the S-SFT and I-GRPO training phases: StrategyQA (Geva et al., 2021), which requires multi-step logical reasoning and integration of scattered information for strategic question answering, and MMLU (Hendrycks et al., 2020), which requires comprehensive cross-domain knowledge and advanced reasoning capabilities for evaluating general language understanding across diverse academic disciplines. Table 4 details the experimental results, highlighting Dubito-Pro’s ability to generalize effectively to unseen tasks.

5 Conclusion

In this work, we introduce Dubito-Pro, a framework that transitions adaptive reasoning from opaque reward maximization to transparent self-awareness. By identifying Entropy Variance as a robust signal of cognitive instability, we enable LLMs to dynamically allocate compute based on intrinsic need rather than static heuristics. To internalize this capability, we propose Intrinsic-Weighted GRPO (I-GRPO), a novel RL algorithm that incentivizes the alignment of reasoning strategy with internal uncertainty. Dubito-Pro establishes a superior Pareto frontier, reducing token usage by 80% on simple tasks and improving overall accuracy by 7.47%. We conclude that robust efficiency requires models to perceive their own limitations, proving that for AI, the ability to doubt is the precursor to efficient and trustworthy existence.

Limitations

While Dubito-Pro establishes a new paradigm for interpretable and efficient adaptive reasoning, we acknowledge several limitations inherent to our mechanism-driven approach :

Subjective vs. Objective Uncertainty (The "False Confidence" Problem). The most significant limitation of Dubito-Pro is its reliance on the model's intrinsic perception of uncertainty. Our Entropy Variance metric (σ_t) captures subjective uncertainty (the model knowing it is confused) but cannot detect objective error when the model is confident in a falsehood. For instance, if the model holds a strong but incorrect belief (e.g., a hallucinated historical fact or a fundamental misconception of physics), the predictive distribution remains sharp (low entropy/variance), and the Slow Thinking mode is not triggered. Thus, Dubito-Pro is highly effective against logical instability but less effective against knowledge boundaries or confident hallucinations.

Binary Granularity. Currently, Dubito-Pro switches between two discrete policies: Fast and Slow. However, human cognition operates on a continuous spectrum. A binary switch may be sub-optimal for "intermediate" difficulty tasks that require more than a simple assertion but less than a full-page proof. The strict dichotomy might lead to slight over-spending on medium-difficulty tasks where a "Medium" reasoning mode would have sufficed.

Inference Overhead of Monitoring. Although calculating entropy variance is computationally cheaper than generating tokens, it requires access to the full vocabulary logits at every decoding step. In highly optimized inference engines (e.g., vLLM, TGI) that aggressively prune or quantize logits for speed, extracting the full distribution to compute normalized entropy introduces a non-negligible latency overhead. While this cost is offset by the reduction in generated tokens, it poses an engineering challenge for ultra-low-latency deployment.

Ethical Considerations

Our work promotes environmental sustainability by significantly reducing inference compute, aligning with Green AI principles; however, users must guard against automation bias, as our intrinsic uncertainty metric captures subjective confusion

rather than objective falsity, potentially allowing confident hallucinations to bypass verification. Additionally, while incentivizing efficiency could theoretically reinforce heuristic biases inherent in "Fast Thinking," Dubito-Pro mitigates this by providing a transparent, auditable signal of cognitive struggle, offering a mechanism to detect and debug reasoning failures in safety-critical applications.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. 2025. The unreasonable effectiveness of entropy minimization in llm reasoning. *arXiv preprint arXiv:2505.15134*.
- Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.
- Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. 2024. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.
- Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *arXiv preprint arXiv:2508.15260*.
- Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

- Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, YK Li, et al. 2024. Deepseek-coder: When the large language model meets programming—the rise of code intelligence. *arXiv preprint arXiv:2401.14196*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Chen Huang, Wei Lu, and Wenxuan Zhang. 2025. Pear: Phase entropy aware reward for efficient reasoning. *arXiv preprint arXiv:2510.08026*.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.
- Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*.
- Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. 2025. Learn to reason efficiently with adaptive length-based reward shaping. *arXiv preprint arXiv:2505.15612*.
- Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025a. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*.
- Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. 2025b. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. <https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1.5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2>. Notion Blog.
- Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.
- Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, M Waleed Kadous, and Ion Stoica. 2024. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*.
- Ziqing Qiao, Yongheng Deng, Jiali Zeng, Dong Wang, Lai Wei, Guanbo Wang, Fandong Meng, Jie Zhou, Ju Ren, and Yaoxue Zhang. 2025. Concise: Confidence-guided compression in step-by-step efficient reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 8021–8040.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.
- Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.
- Artem Vazhentsev, Lyudmila Rvanova, Ivan Lazichny, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2025. Token-level density-based uncertainty quantification methods for eliciting truthfulness of large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2246–2262.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Michelle Wastl, Jannis Vamvas, and Rico Sennrich. 2025. Uzh at semeval-2025 task 3: Token-level self-consistency for hallucination detection. In *Proceed-*

ings of the 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 257–270.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhiqiu Xia, Jinxuan Xu, Yuqian Zhang, and Hang Liu. 2025. A survey of uncertainty estimation methods on large language models. *arXiv preprint arXiv:2503.00172*.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

William J Youden. 1950. Index for rating diagnostic tests. *Cancer*, 3(1):32–35.

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

Yujian Zhang, Keyu Chen, Zhifeng Shen, Ruizhi Qiao, and Xing Sun. 2025b. Adaptive dual reasoner: Large reasoning models can think efficiently by hybrid reasoning. *arXiv preprint arXiv:2510.10207*.

A Data Construction & Prompt Engineering

To enable the model to master two distinct reasoning policies—Fast Thinking (π_{fast}) and Slow Thinking (π_{slow})—we constructed a high-quality paired dataset using GPT-5 as the teacher model. This appendix details the prompting strategies, filtering heuristics, and statistical properties of the resulting dataset.

A.1 Teacher Model Prompts

We employed specific system instructions to enforce strict stylistic and formatting constraints. The core requirement was that both modes must yield the correct answer but via fundamentally different cognitive trajectories.

Fast Thinking Prompt. The goal of this prompt is to elicit a direct, conclusion-oriented reasoning path that skips unnecessary verification steps, simulating "System 1" efficiency.

Slow Thinking Prompt. The goal of this prompt is to elicit a deliberate, self-reflective reasoning path that actively searches for errors, simulating "System 2" rigor.

A.2 Data Filtering & Cleaning

Generating synthetic data at scale requires rigorous quality control to prevent hallucinations or format deviations from contaminating the SFT process. We applied a three-step filtering pipeline:

- Format Validation:** We used regular expressions to verify that every generated response contained a valid `<think>...</think>` block and a `\boxed{...}` answer.
 - Regex: `r"<think>(.*?)</think>.*The final answer is boxed{(.*?)}"`
 - Responses failing this pattern were discarded (approx. 2% rejection rate).
- Correctness Verification:** We extracted the content within `\boxed{}` and compared it against the ground truth provided in the source datasets (GSM8K, DeepScaleR, etc.). To handle formatting differences (e.g., "1,000" vs "1000", or "7.0" vs "7"), we utilized a symbolic math equivalence checker (SymPy) for math problems and exact string matching/normalization for multiple-choice QA.
 - Pairs where either the Fast or Slow response was incorrect were entirely dis-

Fast Thinking Prompt

[System Prompt]:

You are an ultra-efficient, speed-focused reasoning engine. Your goal is to solve the user's problem using the absolute minimum number of tokens required to ensure accuracy.

[Style Constraints]:

- Directness:** Do not use filler phrases like "Let's think" or "To solve this." Jump straight into the calculation or logic.
- Conciseness:** Use short sentences and bullet points where possible.
- No Reflection:** Do NOT perform self-correction, double-checking, or exploring alternative methods. Assume your first intuition is correct.

[Output Format]:

`<think>`

[Your concise reasoning steps here]

`</think>`

[Your final result here]

[User Input]:

{Question}

Figure 5: Prompt for generating a direct, conclusion-oriented reasoning path.

Domain	Source Dataset	Count	Description
Mathematics	DeepScaleR	2,862	Multi-step arithmetic and algebra.
Logic	GSM8K	1,194	Implicit reasoning and symbolic manipulation.
Commonsense	CommonsenseQA	1,032	Knowledge retrieval and basic inference.
Total	–	5,088	–

Table 5: Detailed Data Sources and Data Volume Statistics

carded to ensure the model only learns from correct reasoning trajectories.

3. **Style Contrast Enforcement:** To ensure the "Slow" data actually provides more information than the "Fast" data, we applied a length heuristic. We discarded any pair where $\text{Length}(\text{Slow}) < 1.5 * \text{Length}(\text{Fast})$. This ensures that the Slow Thinking mode contains significant additional cognitive work (verification, decomposition) compared to the Fast mode.

A.3 SFT Dataset Statistics

After filtering, the final S-SFT dataset consists of 5,088 pairs (10,176 total samples). The data is sourced from three primary domains to ensure generalization.

Token Length Distribution. The stylistic prompts successfully induced distinct distributions in token length (measured via DeepSeek-R1-Distill-Qwen-7 Tokenizer), as shown in the statistics below:

- **Fast Thinking Mode:**
 - Mean Length: 431 tokens
 - Standard Deviation: 45 tokens
 - Characteristics: Linear, procedural, minimal branching.
- **Slow Thinking Mode:**
 - Mean Length: 1289 tokens
 - Standard Deviation: 210 tokens
 - Characteristics: Contains average of 2.3 "verification" keywords (e.g., "verify", "check", "ensure") per sample.

This clear separation in length and content complexity confirms that the SFT data provides a strong signal for the model to distinguish between the two operating modes.

B Implementation & Hyperparameters

We implement Dubito-Pro using the PyTorch framework, leveraging the Hugging Face transformers ecosystem and the trl (Transformer Reinforcement Learning) library. This section details the architectural choices, training configurations, and computational infrastructure used to obtain our main results.

B.1 Model Architecture

We utilize DeepSeek-R1-Distill-Qwen-7 and Qwen3-8B as our base model, chosen for its robust instruction-following capabilities and high reasoning baseline. To maintain computational efficiency and manage memory constraints during the reinforcement learning phase, we employ Low-Rank Adaptation (LoRA) (Hu et al., 2022) across all training stages rather than full fine-tuning. We apply LoRA adapters to all linear layers (q_{proj} , k_{proj} , v_{proj} , o_{proj} , $gate_{proj}$, up_{proj} , $down_{proj}$) with a rank $r = 64$, $\alpha = 128$, and a dropout rate of 0.05.

B.2 Training Hyperparameters

Our three-stage pipeline requires distinct hyperparameter configurations to balance format learning with policy optimization.

Stage 1: Stylized Supervised Fine-Tuning (S-SFT). The objective of this stage is to instill the distinct "Fast" and "Slow" reasoning formats. We optimize the model using a standard cross-entropy loss for 3 epochs, which we found sufficient for format alignment without inducing overfitting. The training utilizes a global batch size of 128 (achieved via gradient accumulation) and a maximum sequence length of 32768 tokens to accommodate detailed reasoning traces. We employ the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e - 8$) with a learning rate of 2×10^{-5} and a cosine decay

Slow Thinking Prompt

[System Prompt]:

You are a meticulous, cautious, and self-reflective reasoning expert. Your goal is to solve the user's problem with maximum reliability, treating every assumption with skepticism.

[Style Constraints]:

1. **Decomposition:** Start by breaking the problem down into its fundamental components.
2. **Metacognition:** Explicitly state your internal monologue. Use phrases like "Wait, let me double-check that," "Is there an edge case here?" or "Let's verify this calculation."
3. **Verification:** After deriving a tentative answer, you **MUST** perform a verification step (e.g., plugging values back in, checking logical consistency) before concluding.

[Output Format]:

```
<think>
[Your detailed, reflective reasoning steps here]
</think>
[Your final result here]
```

[User Input]:

```
{Question}
```

Figure 6: Prompt for generating a deliberate, self-reflective reasoning path.

schedule following a 5% warmup period. A weight decay of 0.01 is applied for regularization.

Stage 2: M-Calibrate. For the calibration of the entropy variance threshold (τ_σ), we rely on a held-out validation set of 1,000 samples containing mixed difficulty levels. We calculate the entropy standard deviation using a sliding window size of $W = 10$ tokens. By maximizing Youden's J statistic on this validation set to distinguish between correct and incorrect reasoning paths, we determine the optimal switching threshold to be $\tau_\sigma = 0.98$.

Stage 3: Intrinsic-Weighted GRPO (I-GRPO). This stage employs our novel Online RL algorithm, where hyperparameter selection is critical for stability. We set the learning rate significantly lower at 5×10^{-7} to ensure stable policy updates. A key configuration is the Group Size (G) of 16, which enables our Hybrid Sampling strategy to facilitate effective comparative learning. The training runs for one epoch over the dataset. To penalize deviations from the reference SFT policy, we use a KL coefficient (β_{KL}) of 0.04 and a standard PPO clip ratio (ϵ) of 0.2. Crucially, the hyperparameters governing our novel algorithmic contributions are set

as follows: the Cognitive Alignment Weight (β) is set to 0.1 to incentivize consistency between internal state and behavior, and the Advantage Scaling Factor (γ) is set to 1.0, effectively amplifying the learning signal for samples with maximum uncertainty. A slight length penalty ($\lambda = 0.00003$) is included to discourage verbosity loops.

Infrastructure. All training experiments were conducted on a single high-performance compute node equipped with $8 \times$ NVIDIA A100 (80GB SXM4) GPUs. To handle the significant memory requirements of generating large groups of trajectories during I-GRPO, we utilized DeepSpeed ZeRO-3 (with CPU offloading disabled) to partition model states across GPUs. We further accelerated the training process using FlashAttention-2. Under these settings, the S-SFT stage completed in approximately 8 hours, while the more computationally intensive I-GRPO stage required approximately 18 hours due to the overhead of on-policy generation. The software stack included Python 3.10.17, PyTorch 2.8.0, Transformers 4.57.3, and TRL 0.24.0.

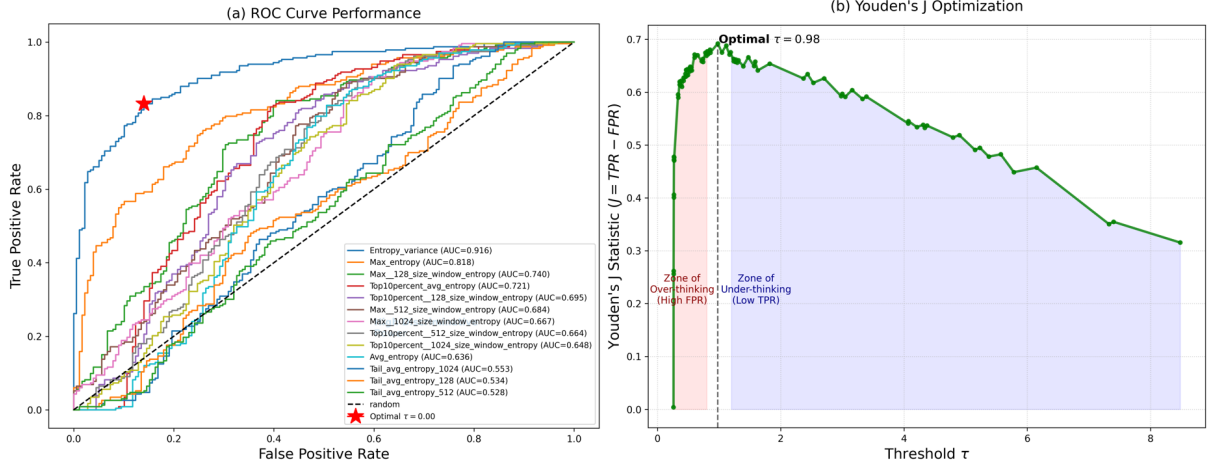


Figure 7: Threshold Calibration Analysis on the Validation Set.

C Additional Experimental Results

This appendix supplements the main experimental findings with a detailed sensitivity analysis of our core hyperparameters, an examination of the I-GRPO training dynamics, and a granular breakdown of performance across individual datasets.

Model	Overall Activation Rate (SAR)	Positive Switch Rate (PSR) \uparrow	Wasted Switch Rate (WSR) \downarrow
Dubito-SFT	11.58	69.71	34.96
Dubito-Pro (DPO)	32.38	75.78	27.36
AdaptThink	43.02	84.82	17.46
Dubito-Pro (I-GRPO)	47.42	88.25	12.16

Table 6: **Comparative analysis of switching efficacy across adaptive methods.** We evaluate the quality of mode transitions on the Mixed-Difficulty Benchmark.

C.1 Adaptive Mechanism Validation

Having established the superior efficiency-accuracy trade-off of Dubito-Pro, we now investigate the internal dynamics driving this performance. Specifically, we seek to validate whether the model’s mode-switching behavior correlates with intrinsic task difficulty and whether these switches meaningfully contribute to error correction.

Activation Rate Analysis. To visualize the model’s sensitivity to problem complexity, we plot the Slow Activation Rate (SAR) across benchmarks of increasing difficulty in Figure 4. The results reveal a distinct, monotonic relationship between task difficulty and the utilization of Slow Thinking. For easy task GSM8K, Dubito-Pro maintains an SAR of only 2.5%, suggesting that for factual retrieval where internal knowledge is stable, negligible Slow Thinking mode. In contrast, for medium-difficulty task MATH500, the SAR rises to 46.4%,

reflecting the dataset’s heterogeneous nature where multi-step logic triggers higher uncertainty variance. On the most challenging task AIME24, the SAR surges to 93.3%, confirming the model’s ability to identify the high cognitive load required for advanced mathematics. Crucially, comparison with AdaptThink reveals significant behavioral differences. AdaptThink’s activation curve is notably flatter, maintaining an SAR of $\sim 18\%$ on easy task and only $\sim 73\%$ on hard task. This indicates that AdaptThink tends to "overthink" simple queries, while "underthinking" complex queries. Dubito-Pro’s variance-based mechanism can filter out these false positives and false negatives, thereby achieving more accurate and efficient resource allocation.

Switching Efficacy. To evaluate the quality of switching decisions, we categorize transitions into two types: positive switches and wasted switches. **Positive switches** occur when the Slow mode successfully resolves a task that the Fast mode failed, quantifying the mechanism’s ability to "rescue" the model from errors. Conversely, **wasted switches** refer to instances where the model unnecessarily triggers the Slow mode despite the Fast mode being sufficient, representing redundant computational overhead. Table 6 presents the switching efficacy analysis. Dubito-Pro achieves the highest Positive Switch Rate (88%) among all adaptive methods. Furthermore, Dubito-Pro demonstrates the lowest Wasted Switch Rate (12%), significantly lower than AdaptThink (17%). We attribute this to the I-GRPO training objective. Dubito-Pro’s Cognitive Alignment Reward explicitly penalizes misalignment between internal confidence and behavioral mode, teaching the model to trust its Fast Thinking when

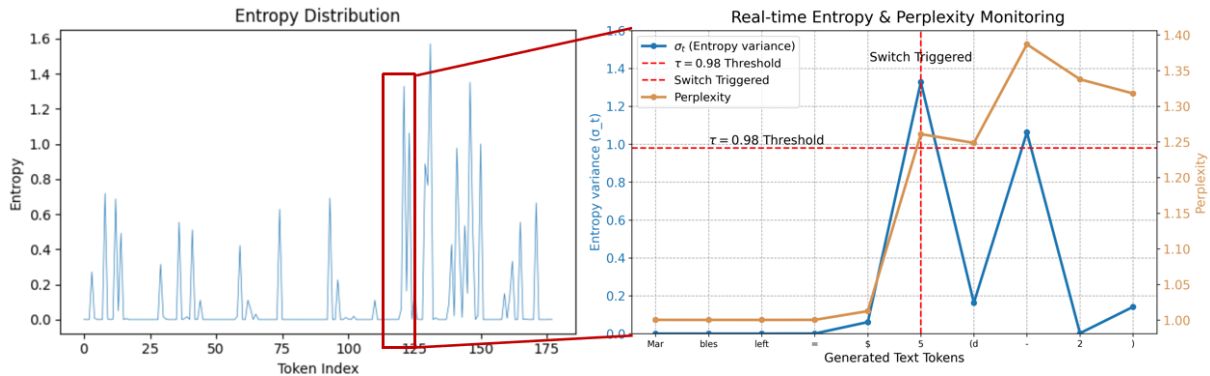


Figure 8: Prompt for eliciting a direct, conclusion-oriented reasoning path.

its internal state is stable. This proves that Dubito-Pro’s adaptation is driven by a genuine perception of cognitive need rather than a learned heuristic for reward maximization.

C.2 Sensitivity Analysis

The effectiveness of Dubito-Pro relies on the robust detection of cognitive instability. Here, we analyze how the two critical hyperparameters of our detection mechanism—the sliding window size (w) and the switching threshold τ —impact model behavior.

Impact of Sliding Window Size (w). We evaluated window sizes ranging from $w = 1, 5, 10, 15, 20$ on the validation set. Our results indicate a clear trade-off between noise sensitivity and detection latency. A small window ($w \leq 5$) results in a high False Positive Rate (FPR), as the metric becomes overly sensitive to benign, momentary spikes in entropy caused by synonyms or open-ended phrasing. Conversely, a large window ($w \geq 15$) excessively smooths the signal, causing the model to miss rapid oscillatory patterns characteristic of degenerative loops or acute logical conflicts. We found that $w = 10$ offers the optimal balance, maximizing the Youden Index by capturing local variance while filtering out transient noise.

Threshold Calibration (τ). Figure 7 would illustrate the Receiver Operating Characteristic (ROC) curve derived during the Stage 2: M-Calibrate phase. We swept the threshold τ from 0.0 to 10.0. The selected threshold of $\tau = 0.98$ corresponds to the "knee" of the curve, representing the point of maximum separation between correct Fast Thinking trajectories (typically stable, low variance) and incorrect ones (high variance). Increasing τ beyond 1.26 leads to a sharp drop in Slow Thinking activa-

tion on Hard tasks (under-thinking), while decreasing it below 0.77 causes near-constant switching on Easy tasks (over-thinking), degrading efficiency.

D Qualitative Case Studies

To provide granular insight into the decision-making process of Dubito-Pro, we present qualitative examples analyzing the model’s token-level generation trajectories.

D.1 Case Studies on Benchmarks

We present case studies of Dubito-Pro in Figures 10–14 to illustrate its capacity for adaptive thinking-mode selection. As shown in Figures 10, 11, and 12, Qwen3-8B incurs significant token overhead on relatively simple problems, often producing unnecessary steps and redundant attempts. In contrast, our Dubito-Pro-8B model adaptively selects the FastThink mode, directly generating concise solutions without superfluous reflection. Notably, this capability holds even when the input problems differ from our training data in format and subject. On the other hand, the cases in Figures 13 and 14 demonstrate that when faced with difficult problems, Dubito-Pro-8B automatically employs the SlowThink mode to ensure accuracy.

D.2 Failure Analysis and Limitations

To maintain rigor, we analyze two distinct scenarios where Dubito-Pro’s mechanism yields suboptimal results, highlighting the distinction between subjective uncertainty and objective truth.

The first failure mode is the "Wasted Switch," often observed in factual retrieval tasks like TriviaQA. For example, when asked "Who is the lead singer of Xylophonic Zephyrs?" (a fictitious or rare entity), the model encounters high lexical uncertainty when attempting to generate the proper noun.

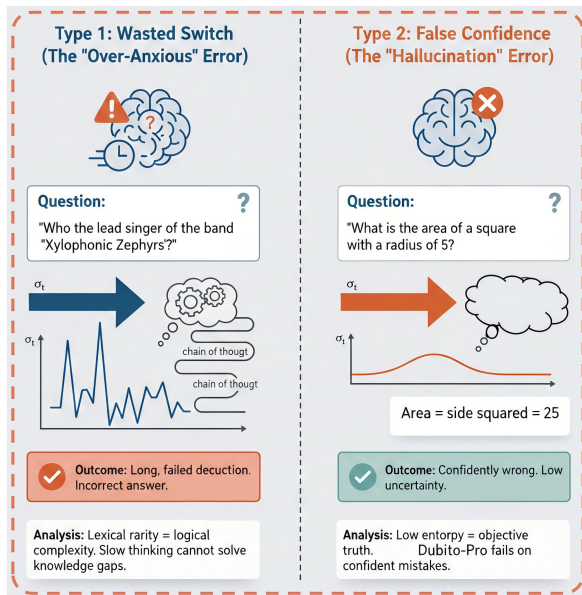


Figure 9: Prompt for eliciting a direct, conclusion-oriented reasoning path.

This causes σ_t to spike, triggering a switch to Slow Thinking. However, since the error stems from a fundamental knowledge gap rather than a lack of reasoning depth, the subsequent detailed reflection cannot recover the correct information. The model expends tokens on a long chain of thought only to hallucinate an answer, reducing the AES for that query.

The second, more dangerous failure mode is "False Confidence." This occurs when the model proceeds with a false premise that it "believes" to be true. For instance, if asked about the "radius of a square," a model might confidently apply the area formula for a circle, conflating the concepts. Because the model is not internally conflicted—it is confidently wrong—the entropy variance remains low, and the switching threshold is never breached. This highlights a fundamental limitation of our approach: Dubito-Pro optimizes based on the model's intrinsic uncertainty, which is a subjective measure. It cannot correct errors where the model's internal world model is fundamentally aligned but factually incorrect. Future work must address this by integrating external verifiers or uncertainty calibration techniques specifically targeting hallucinations.

Question (GSM8K): Dan plants 3 rose bushes. Each rose bush has 25 roses. Each rose has 8 thorns. How many thorns are there total?

Qwen3-8B: (747 Tokens)

<think>Okay, let's see. Dan plants 3 rose bushes. Each rose bush has 25 roses. Each rose has 8 thorns. How many thorns are there total? Hmm, [...] (omit 184 tokens) Hmm, 75 times 8. Let me think. 70 times 8 is 560, and 5 times 8 is 40. Adding those together, 560 + 40 equals 600. So, 75 times 8 is 600. **Therefore, there are 600 thorns in total.** Wait, let me check that again to make sure I didn't make a mistake. So, 3 bushes, 25 roses each: $3 \times 25 = 75$. Then 75 roses, 8 thorns each: 75×8 . Let me do 75×8 another way. 75×2 is 150, so 75×8 is 150×4 , which is 600. **Yeah, that still gives me 600.** Okay, that seems right. Alternatively, maybe I can think of it as multiplying all the numbers together. So, 3 rose bushes * 25 roses per bush * 8 thorns per rose. **So, 3×25 is 75, then 75×8 is 600.** Yep, same result. I don't think there's any trick here, like some roses not having thorns or something. The problem says each rose has 8 thorns, so all of them contribute. So, multiplying them all together should give the total number of thorns. Another way to check: Let's break down 75×8 . **75 is $70 + 5$. 70×8 is 560, 5×8 is 40. $560 + 40$ is 600.** Yep. So that's consistent. Alternatively, using another multiplication method. 8×75 . **If I think of 8×75 as $8 \times (70 + 5) = 8 \times 70 + 8 \times 5 = 560 + 40 = 600$.** Same answer. So, all the different ways of calculating it lead me to 600. Therefore, I think the total number of thorns is 600. I don't see any other factors here. The problem is straightforward: multiply the number of bushes by roses per bush, then multiply by thorns per rose. No need to worry about any exceptions or anything. So, yeah, 600 is the answer. ****Final Answer****The total number of thorns is `\boxed{600}`.</think>To determine the total number of thorns, we need to follow a step-by-step multiplication process based on the information provided:---### Step 1: Calculate the total number of rosesDan plants **3 rose bushes**, and each bush has **25 roses**. So, the total number of roses is:
$$3 \times 25 = 75$$
---### Step 2: Calculate the total number of thornsEach rose has **8 thorns**, so we multiply the total number of roses by the number of thorns per rose:
$$75 \times 8 = 600$$
---### Final Answer
$$\boxed{600}$$

Dubito-Pro-8B: (205 Tokens)

<think>Let me break this problem down step by step, as a tutor would when guiding a student through a multi-layered multiplication word problem. First, Dan plants 3 rose bushes. That's our starting quantity. Each rose bush has 25 roses. So, to find the total number of roses, I multiply the number of bushes by the number of roses per bush: $3 \text{ bushes} \times 25 \text{ roses per bush} = 75 \text{ roses}$. Now, each rose has 8 thorns. So, to find the total number of thorns, I multiply the total number of roses by the number of thorns per rose: $75 \text{ roses} \times 8 \text{ thorns per rose}$. Let me compute that: $75 \times 8 = (70 \times 8) + (5 \times 8) = 560 + 40 = 600$. **So, the total number of thorns is 600.**</think>The final answer is `\boxed{600}`

Figure 10: Comparison on a simple GSM8K task: Qwen3-8B wastes computation with 747 reasoning tokens containing redundant steps, whereas Dubito-Pro-8B adaptively activates FastThink mode to generate a concise solution.

Question (MATH500): Simplify $3t+4-6t+7t-4$.

Qwen3-8B: (1891 Tokens)

<think>Okay, so I need to simplify this algebraic expression: $3t + 4 - 6t + 7t - 4$. Let me think about how to approach this. First, I remember that simplifying an expression like this involves combining like terms. Like terms are terms that have the same variable part, right? So, terms with 't' are like terms, and constant terms (numbers without variables) are also like terms. Let me write down the expression again to visualize it better: $3t + 4 - 6t + 7t - 4$. So, let me separate the terms into those with 't' and those without. The terms with 't' are $3t$, $-6t$, and $7t$. The constant terms are $+4$ and -4 . Starting with the 't' terms: $3t - 6t + 7t$. Let me compute that step by step. First, $3t$ minus $6t$. Hmm, $3t - 6t$ is like $3 - 6$ which is -3 , so that would be $-3t$. Then, adding $7t$ to that. So, $-3t + 7t$. That would be $(-3 + 7)t$, which is $4t$. Okay, so combining those gives me $4t$. Now, the constant terms: $+4 - 4$. That's straightforward. $4 - 4$ is 0 . So those two terms cancel each other out. So putting it all together, the simplified expression is $4t + 0$, which is just $4t$. Wait, [...] (omit 1144 tokens) Which is same as before. Alternatively, maybe I can factor something out? But I don't think that's necessary here. Alternatively, think of the entire expression as $(3t - 6t + 7t) + (4 - 4)$. Which is again $4t + 0$. So yeah, $4t$. I think that's the answer. But just to make sure, let me think if there's any other way this could be interpreted. For example, could there be a different grouping? Like, if there were parentheses, but there aren't any. The original expression is written as $3t + 4 - 6t + 7t - 4$. So order of operations is just left to right, and all the operations are addition and subtraction. So, no ambiguity in the order. So, combining like terms is the correct approach. Therefore, I think the simplified expression is $4t$. **Final Answer**The simplified expression is $\boxed{4t}$.

</think>To simplify the expression: $3t + 4 - 6t + 7t - 4$ we follow the standard process of combining like terms. This involves grouping and simplifying terms that share the same variable or are constants. ---### Step 1: Group Like TermsWe identify two types of terms: - **Terms with the variable t **: $3t$, $-6t$, $7t$ - **Constant terms**: $+4$, -4 ---### Step 2: Combine the t Terms $3t - 6t + 7t = (3 - 6 + 7)t = 4t$ ---### Step 3: Combine the Constant Terms $4 - 4 = 0$ ---### Step 4: Combine ResultsNow, combine the simplified t terms and the constant terms: $4t + 0 = 4t$ ---### Final Answer $\boxed{4t}$

Dubito-Pro-8B: (177 Tokens)

Let me begin by carefully simplifying the expression: $3t + 4 - 6t + 7t - 4$ First, I will identify and group all the like terms. The terms with t are: $3t$, $-6t$, and $7t$. The constant terms are: $+4$ and -4 . Now, let me combine the t terms: $3t - 6t + 7t = (3 - 6 + 7)t = 4t$ Next, combine the constants: $4 - 4 = 0$ So the entire expression simplifies to: $4t + 0 = 4t$ Final answer: $\boxed{4t}$

</think>The final answer is $\boxed{4t}$

Figure 11: Performance on a MATH500 problem: Qwen3-8B requires 1,891 reasoning tokens, whereas Dubito-Pro-8B leverages FastThink mode to solve the problem concisely in only 177 tokens.

Question (AIME24): Every morning, Aya does a 9 kilometer walk, and then finishes at the coffee shop. One day, she walks at s kilometers per hour, and the walk takes 4 hours, including t minutes at the coffee shop. Another morning, she walks at $s+2$ kilometers per hour, and the walk takes 2 hours and 24 minutes, including t minutes at the coffee shop. This morning, if she walks at $s+\frac{1}{2}$ kilometers per hour, how many minutes will the walk take, including the t minutes at the coffee shop?

Qwen3-8B: (5134 Tokens)

`<think>[...] (omit 4014 tokens) </think>` To solve the problem, we analyze the two given scenarios involving Aya's walk and the time she spends at the coffee shop. The key is to determine the relationship between her walking speed, the time spent walking, and the time spent at the coffee shop. ---### Step 1: Define Variables Let: s = Aya's walking speed in km/h t = time spent at the coffee shop in minutes $x = \frac{t}{60}$ = time spent at the coffee shop in hours ---### Step 2: Use the Distance Formula The distance Aya walks is always 9 km. Using the formula: $\text{Distance} = \text{Speed} \times \text{Time}$ We can write two equations based on the two scenarios: ---### First Day: Speed: s km/h Total time: 4 hours - Walking time: $4 - x$ hours - Equation: $9 = s(4 - x)$ ---### Second Day: Speed: $s + 2$ km/h Total time: 2 hours 24 minutes = 2.4 hours - Walking time: $2.4 - x$ hours - Equation: $9 = (s + 2)(2.4 - x)$ ---### Step 3: Solve the System of Equations From the first equation: $s(4 - x) = 9 \rightarrow s = \frac{9}{4 - x}$ Substitute into the second equation: $9 = \left(\frac{9}{4 - x} + 2\right)(2.4 - x)$ Simplify: $9 = \left(\frac{9 + 2(4 - x)}{4 - x}\right)(2.4 - x)$ $9 = \left(\frac{9 + 8 - 2x}{4 - x}\right)(2.4 - x)$ $9 = \left(\frac{17 - 2x}{4 - x}\right)(2.4 - x)$ Multiply both sides by $4 - x$: $9(4 - x) = (17 - 2x)(2.4 - x)$ Expand both sides: - Left: $36 - 9x$ - Right: $17 \cdot 2.4 - 17x - 2x \cdot 2.4 + 2x^2 = 40.8 - 17x - 4.8x + 2x^2 = 40.8 - 21.8x + 2x^2$ Set equal: $36 - 9x = 40.8 - 21.8x + 2x^2$ Bring all terms to one side: $2x^2 - 12.8x + 4.8 = 0$ Divide by 2: $x^2 - 6.4x + 2.4 = 0$ Solve using the quadratic formula: $x = \frac{6.4 \pm \sqrt{6.4^2 - 4 \cdot 1 \cdot 2.4}}{2}$ $x = \frac{6.4 \pm \sqrt{40.96 - 9.6}}{2} = \frac{6.4 \pm \sqrt{31.36}}{2}$ $x = \frac{6.4 \pm 5.6}{2}$ Solutions: $x = \frac{12}{2} = 6$ or $x = \frac{0.8}{2} = 0.4$ Since $x = \frac{t}{60}$, and t must be less than the total time (which is 4 hours = 240 minutes), we discard $x = 6$ (which would imply $t = 360$ minutes). So: $x = 0.4 \rightarrow t = 0.4 \times 60 = 24$ minutes Now find s : $s = \frac{9}{4 - x} = \frac{9}{4 - 0.4} = \frac{9}{3.6} = 2.5$ km/h ---### Step 4: Calculate the Third Day's Total Time On the third day, Aya walks at: $s + \frac{1}{2} = 2.5 + 0.5 = 3.0$ km/h Walking time: $\frac{9}{3.0} = 3.0$ hours = 180 minutes Total time (including 24 minutes at the coffee shop): $180 + 24 = 204$ minutes ---### Final Answer: $\boxed{204}$

Dubito-Pro-8B: (1694 Tokens)

`<think>` Let me carefully break this problem down step by step, with full metacognitive reflection. First, [...] (omit 955 tokens) Now solve this quadratic using the quadratic formula: $s = \frac{-8 \pm \sqrt{8^2 - 4 \cdot 4 \cdot (-45)}}{2 \cdot 4}$ Compute discriminant: $64 + 720 = 784$ $\sqrt{784} = 28$ So: $s = \frac{-8 \pm 28}{8}$ We discard the negative root because speed can't be negative: $s = \frac{20}{8} = 2.5$ So $s = 2.5$ km/h Now, let's find t . Use equation 1: $\frac{9}{s} + \frac{t}{60} = 4$ Plug in $s = 2.5$: $\frac{9}{2.5} + \frac{t}{60} = 4$ So: $3.6 + \frac{t}{60} = 4$ Then: $\frac{t}{60} = 0.4$ Multiply both sides by 60: $t = 24$ minutes Let me double-check this with the second equation to be sure. Second equation: $\frac{9}{s+2} + \frac{t}{60} = 2$ $\frac{9}{4.5} + \frac{24}{60} = 2$ $\frac{9}{4.5} = 2$ $\frac{24}{60} = 0.4$ So total = $2 + 0.4 = 2.4$ hours - matches! Perfect. Now, the question: This morning, she walks at $s + \frac{1}{2} = 2.5 + 0.5 = 3.0$ km/h. We need the total walk time, including $t = 24$ minutes at the coffee shop. So: Walking time = $\frac{9}{3.0} = 3.0$ hours Coffee shop = $\frac{24}{60} = 0.4$ hours Total time = $3.0 + 0.4 = 3.4$ hours Convert to minutes: $3.4 \times 60 = 204$ minutes Let me verify: 3 hours = 180 minutes 0.4 hours = 24 minutes Total = $180 + 24 = 204$ minutes This seems correct because: - At 3 km/h, 9 km takes 3 hours - correct. - Coffee shop is 24 minutes - same as before. - Total: 3h24m = 204 minutes. Final check: Did I misread the question? It says "how many minutes will the walk take, including the t minutes at the coffee shop?" - yes, 204. My initial thought was to set up the equations, and I double-checked everything. The logic holds. This seems correct because: - $s = 2.5$ confirmed by both equations. - $t = 24$ confirmed. - New speed = 3.0, walking time = 3h, total = 3h24m = 204 min. All steps are consistent and verified. `</think>` The final answer is $\boxed{204}$

Figure 12: On challenging AIME 2024 tasks, Dubito-Pro-8B utilizes the SlowThink mode to solve the problem, avoiding premature direct generation.

Question(StrategyQA): Is the language used in Saint Vincent and the Grenadines rooted in English?

Qwen3-8B: (397 Tokens)

<think>Okay, let's tackle this question. The user is asking if the language used in Saint Vincent and the Grenadines is rooted in English. First, I need to recall what I know about the languages of the Caribbean islands, specifically Saint Vincent and the Grenadines. I remember that many Caribbean nations have English as an official language, especially those that were former British colonies. Saint Vincent and the Grenadines is a country in the Caribbean, and I think it was a British colony. So, it's likely that English is one of the official languages there. But wait, the question is about the language being "rooted in English." That could mean that the primary language is English, or that the language has English influences. Let me think. The official languages of Saint Vincent and the Grenadines are English and Vincentian Creole. So, English is definitely an official language, and the creole is based on English with influences from other languages like French, Spanish, and indigenous languages. So, the answer would be yes because the primary language is English, and the creole is rooted in English. But I should double-check if there are any other languages or if there's a different structure. Also, considering that the country is part of the Commonwealth, which often uses English as a lingua franca, that supports the idea that English is foundational. Another angle: even if there's a creole, the root language is English. So the answer should be yes. I don't think there's any other major language that's the root here. So the answer is yes.</think>

A: Saint Vincent and the Grenadines has English as one of its official languages, and the country's primary language is English. While Vincentian Creole is also widely spoken, it is rooted in English with influences from other languages. Thus, the language used in Saint Vincent and the Grenadines is rooted in English. So the answer is **yes**.

Dubito-Pro-8B: (275 Tokens)

<think>Let me break this down carefully. Saint Vincent and the Grenadines is a Caribbean nation that gained independence from the United Kingdom in 1979. As a former British colony, its official language is English. This is standard for most Caribbean nations that were part of the British Empire, such as Jamaica, Barbados, and Trinidad and Tobago. The language of the country is not rooted in English in the sense of being a separate, indigenous language — rather, English is the official and primary language, and it is the language of government, education, and commerce. The people of Saint Vincent and the Grenadines speak English as their first language, and it is deeply embedded in their cultural and administrative systems. There is no indication of a distinct, indigenous language being the root; in fact, the indigenous peoples of the region (such as the Caribs) were largely displaced or assimilated, and English became the dominant language. Therefore, the statement "Is the language used in Saint Vincent and the Grenadines rooted in English?" is asking whether English is the foundational language — and the answer is yes. This seems correct because: - Official language: English - Education, government, media: English - No indigenous language is preserved as the primary language - Historical context: British colonization So the answer is yes.</think>The final answer is $\boxed{\text{yes}}$

Figure 13: Generalization on out-of-distribution StrategyQA: Dubito-Pro-8B successfully activates FastThink for simple problems.

Question (MMLU): To prevent desiccation and injury, the embryos of terrestrial vertebrates are encased within a fluid secreted by the A) amnion B) chorion C) allantois D) yolk sac

Qwen3-8B: (734 Tokens)

<think> Okay, let's tackle this question. The question is about which structure in terrestrial vertebrates encases the embryos to prevent desiccation and injury. The options are amnion, chorion, allantois, or yolk sac. [...] (omit 366 tokens) Let me double-check. The amnion secretes the amniotic fluid, which is the fluid that the embryo is encased in. That fluid would prevent desiccation by keeping the embryo moist and also protect it from physical shocks. So the answer should be A) amnion. The other options don't fit the description as well. The chorion is part of the outer membrane, maybe the shell membrane, but not the fluid. The allantois is for waste, and the yolk sac is for nutrients. So yeah, A is the right choice. </think> The question asks which structure encases embryos in terrestrial vertebrates to prevent desiccation and injury. Terrestrial vertebrates, such as reptiles, birds, and mammals, develop in amniotic eggs, which have specialized membranes. The **amnion** is a critical structure in this context. It secretes **amniotic fluid**, which forms a protective, fluid-filled sac around the embryo. This fluid serves two primary functions: (1) it prevents desiccation by maintaining a moist environment, and (2) it cushions the embryo, reducing mechanical injury. The **chorion** is part of the outer egg membrane and facilitates gas exchange, not fluid secretion. The **allantois** is involved in waste storage and gas exchange, while the **yolk sac** provides nutrients. None of these structures directly encase the embryo in a fluid environment. Thus, the correct answer is **A) amnion**. ANSWER: A) amnion. ANSWER: A

Dubito-Pro-8B: (374 Tokens)

<think> Let me break this down step by step. First, I need to recall the structure and function of the extraembryonic membranes in terrestrial vertebrates. These membranes protect the developing embryo and help with gas exchange, waste removal, and preventing desiccation. The four membranes involved are: the amnion, chorion, allantois, and yolk sac. The amnion is the innermost membrane that surrounds the embryo and contains the amniotic fluid. This fluid is crucial for cushioning the embryo, allowing movement, and preventing desiccation by providing a moist environment. So if the question is asking which structure secretes the fluid that encases the embryo to prevent drying out, the amnion is the key. The chorion is the outermost membrane that forms part of the placenta in mammals and facilitates gas exchange, but it doesn't secrete the amniotic fluid. The allantois is involved in waste storage (urine) and in some species helps with respiration, but it doesn't secrete the protective fluid. The yolk sac provides nutrients in early development and is more prominent in reptiles and birds, but again, it doesn't secrete the amniotic fluid. So, the amnion secretes the amniotic fluid that encases the embryo, preventing desiccation and injury. This matches the question exactly. Let me double-check: Yes, in all terrestrial vertebrates (reptiles, birds, mammals), the amnion is responsible for the amniotic fluid. The chorion is the outer membrane, and the allantois is a separate structure. The yolk sac is for nutrition, not fluid secretion. Therefore, the correct answer is A) amnion. </think> ANSWER: A

Figure 14: On challenging MMLU problems, Dubito-Pro-8B appropriately triggers SlowThink yet refrains from over-thinking.