METATHINK: LEARNING WHEN TO THINK VIA MULTI-TOKEN METACOGNITIVE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large reasoning models achieve strong performance through chain-of-thought (CoT) reasoning, but uniformly applying lengthy deliberation to all problems incurs substantial computational overhead. Existing adaptive reasoning methods train models to selectively engage CoT, yet some rely on expensive and inconsistent external annotations, and they typically use a single-token decision mechanism that creates an information bottleneck and leads to decision boundary collapse. Inspired by metacognitive gating in human dual-process theory, we propose METATHINK, which introduces a multi-token metacognitive reasoning stage where the model generates a brief natural language self-assessment before deciding whether to engage CoT. Combined with an asymmetric reward that assigns higher rewards to correct non-CoT responses, METATHINK enables the model to autonomously discover which problems require reasoning through reinforcement learning, without external difficulty annotations. Experiments with DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B on GSM8K, MATH500, and AIME 2024 show that METATHINK improves accuracy–efficiency tradeoffs over existing adaptive reasoning methods across both model scales.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has emerged as a powerful paradigm for training large reasoning models (LRMs) (Guo et al., 2025; OpenAI, 2024; Team, 2025; Snell et al., 2025; Zhao et al., 2024; Hu et al., 2026). By incentivizing chain-of-thought (CoT) reasoning (Wei et al., 2022), these models achieve remarkable performance on challenging benchmarks in mathematics, coding, and scientific reasoning, unlocking capabilities such as self-verification, backtracking, and multi-step planning (Marjanović et al., 2025).

However, LRMs generate hundreds or thousands of thinking tokens regardless of problem difficulty, applying the same exhaustive deliberation to both competition-level problems and straightforward arithmetic (Ma et al., 2025). This uniform resource allocation leads to excessive latency and energy consumption, making inference efficiency a central concern for practical deployment.

To address this, recent work has explored *adaptive reasoning*, which trains models to selectively engage CoT based on problem difficulty (Lou et al., 2025; Zhang et al., 2025a). A common characteristic shared by these methods is that they encode the CoT decision as a single token immediately after the `<think>` tag, reducing the reasoning mode selection to a binary choice at a single position. Under this paradigm, two training strategies have been proposed. The first uses external LLMs to annotate each training problem with a difficulty label, then penalizes the model for deviating from these annotations (Lou et al., 2025). The second directly optimizes an RL objective that encourages the model to skip CoT whenever doing so does not degrade accuracy (Zhang et al., 2025a).

However, these approaches suffer from several limitations. **First**, annotation-based methods depend on external LLMs to determine problem difficulty, but this annotation is inherently unreliable. As we show in §3.1, five frontier LLMs achieve only moderate inter-annotator agreement (Fleiss' $\kappa = 0.594$, where $\kappa = 1$ indicates perfect agreement) on the same problems, with 33.2% of problems receiving conflicting labels across annotators. **Second**, the single-token decision may create an information bottleneck, as the model must compress its difficulty assessment into one discrete token. **Third**, this

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

mechanism is prone to *decision boundary collapse*, where the model converges to always choosing one mode rather than learning a meaningful difficulty-based policy (Lou et al., 2025).

Based on these observations, we draw inspiration from human meta-reasoning in cognitive psychology. According to dual-process theory (Evans & Stanovich, 2013; Kahneman, 2011), humans first generate a rapid intuitive response (System 1), accompanied by a metacognitive *Feeling of Rightness* (Thompson et al., 2011) that gates whether deliberate reasoning (System 2) is triggered (Ackerman & Thompson, 2017). Crucially, this gating involves a rich internal assessment rather than a single binary switch.

We operationalize this mechanism and propose METATHINK, an adaptive reasoning framework with two key innovations. First, we introduce a `<meta>` reasoning stage where the model generates a brief natural language self-assessment before deciding whether step-by-step reasoning is needed, providing a richer decision space than single-token gating. Second, we design an asymmetric reward that assigns higher reward to correct non-CoT responses, enabling the model to autonomously discover which problems it can solve directly through RLVR, without any external difficulty annotations.

Our contributions are summarized as follows:

- We propose METATHINK, an adaptive reasoning framework that introduces a multi-token metacognitive reasoning stage for richer CoT decisions, combined with an asymmetric reward design that eliminates the need for external difficulty annotations.
- We demonstrate that our mechanism can be integrated into standard RLVR training pipelines with minimal modification, making it compatible with most existing reasoning model training frameworks.
- We evaluate METATHINK on GSM8K, MATH500, and AIME using R1-distilled models at two scales, demonstrating better accuracy–efficiency tradeoffs than existing adaptive reasoning methods.

2 RELATED WORK

2.1 LARGE REASONING MODELS AND RLVR

Chain-of-thought (CoT) prompting (Wei et al., 2022) demonstrated that step-by-step reasoning substantially improves LLM performance on complex tasks. Subsequent work further enhanced reasoning through self-consistency (Wang et al., 2022), bootstrapping from correct solutions (Zelikman et al., 2022), and process-level verification (Lightman et al., 2024).

Reinforcement learning with verifiable rewards (RLVR) has since become the dominant paradigm for training large reasoning models (Pan et al., 2025; Snell et al., 2025). DeepSeek-R1 (Guo et al., 2025) showed that pure RL training, using Group Relative Policy Optimization (GRPO) (Shao et al., 2024), can incentivize advanced reasoning behaviors without supervised demonstrations. This approach has been adopted and validated by a series of reasoning models including Kimi k1.5 (Team et al., 2025), QwQ (Team, 2025), OpenAI o1 (OpenAI, 2024), Marco-o1 (Zhao et al., 2024), and Open-Reasoner-Zero (Hu et al., 2026). However, these models generate extended thinking traces for all inputs regardless of difficulty (Marjanović et al., 2025), motivating work on adaptive reasoning.

2.2 ADAPTIVE AND EFFICIENT REASONING

The computational overhead of uniformly long reasoning has motivated a growing body of work on efficient and adaptive reasoning (Feng et al., 2025; Sui et al., 2025). One line of work controls the total reasoning budget at inference time through budget forcing (Muennighoff et al., 2025), token-budget-aware decoding (Han et al., 2025), control tokens (Wen et al., 2025), or certainty-based routing (Lu et al., 2025). Another line compresses or prunes existing reasoning traces to reduce length while preserving accuracy (Hou et al., 2025; Hassid et al., 2025). Other work trains models to adaptively allocate reasoning budgets of varying length (Yu et al., 2025; Zhang et al., 2025b).

Recent methods learn *whether* to engage CoT for each problem. AdaCoT (Lou et al., 2025) uses external LLM difficulty annotations to learn CoT triggering. AdaptThink (Zhang et al., 2025a) uses annotation-free, constrained RL to skip CoT while maintaining accuracy. AutoThink (Tu et al., 2026)

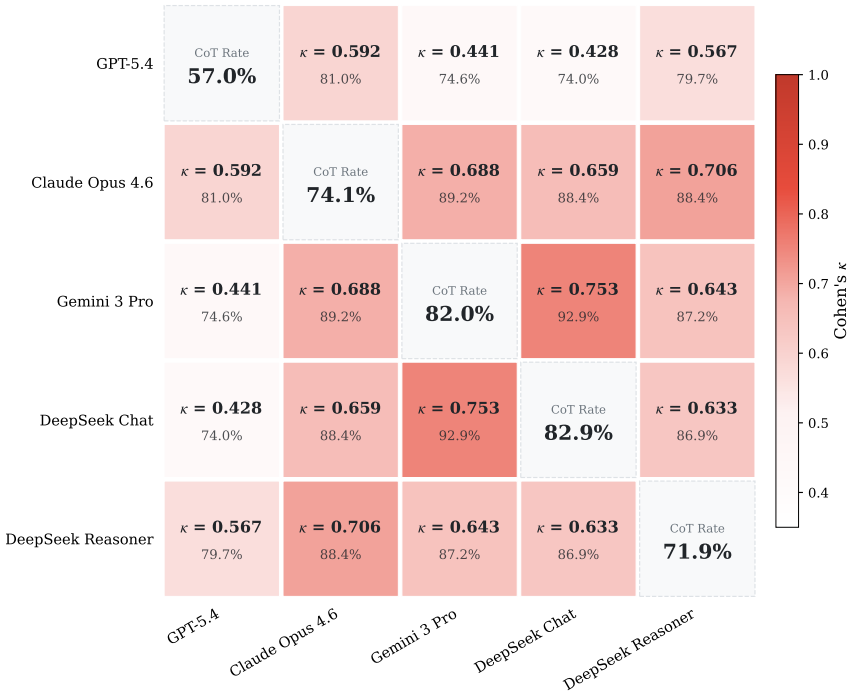


Figure 1: Pairwise annotation consistency (Cohen's κ) among five frontier LLMs on 3,940 math problems. Overall Fleiss' $\kappa = 0.594$.

learns adaptive thinking through multi-stage RL and progressive reward shaping. Ma et al. (2025) showed that NoThinking can match or outperform Thinking under low budgets. These methods typically encode the CoT decision as a single token immediately after the `<think>` tag, creating an information bottleneck prone to decision boundary collapse. Our work introduces a multi-token metacognitive reasoning stage for richer difficulty assessment, with an asymmetric reward that incentivizes efficient reasoning without external annotations.

3 METHOD

3.1 PRELIMINARY

Annotation-driven adaptive reasoning methods such as AdaCoT (Lou et al., 2025) rely on labeling each training sample with whether CoT is necessary. A critical assumption behind this approach is that such labels can be reliably produced. To test this assumption, we conduct a study where five frontier LLMs (GPT-5.4 (OpenAI, 2026), Claude Opus 4.6 (Anthropic, 2026), Gemini 3 Pro (Google DeepMind, 2025), DeepSeek Chat, and DeepSeek Reasoner (Liu et al., 2025)) independently annotate 3,940 math problems from DeepScaleR (Luo et al., 2025) with a binary label indicating whether each problem requires CoT reasoning. As shown in Figure 1, the inter-annotator agreement is only moderate, with an overall Fleiss' $\kappa = 0.594$ (a standard measure of multi-rater agreement, where $\kappa = 1$ indicates perfect agreement and $\kappa < 0.6$ is generally considered moderate). The CoT annotation rates vary dramatically across models, ranging from 57.0% (GPT-5.4) to 82.9% (DeepSeek Chat), and 33.2% of problems receive disagreeing labels across the five annotators. These results demonstrate that the notion of whether a problem "needs" CoT is inherently model-dependent and cannot be determined reliably by external annotation, motivating our annotation-free approach.

Beyond annotation quality, existing methods also rely on a single token to decide whether to engage CoT, which creates a narrow information bottleneck. As we demonstrate in our main experiments (Table 1 and Figure 3), expanding the decision budget to a multi-token metacognitive reasoning stage

CoT mode (complex problem)

```
<think>
<meta>This requires composing f
with
itself and solving.  I should
think
step-by-step.</meta>
f(sqrt(2)) = 2a - sqrt(2)
f(f(sqrt(2))) = a(2a-sqrt(2))^2 -
sqrt(2)
Setting equal to -sqrt(2):
a(2a-sqrt(2))^2 = 0
Since a>0, 2a=sqrt(2), so
a=sqrt(2)/2
</think>
Therefore, a = sqrt(2)/2.
The answer is \boxed{sqrt(2)/2}.
```

Non-CoT mode (simple problem)

```
<think>
<meta>This is a straightforward
sum.
I should not think
step-by-step.</meta>
</think>
991+993+995+997+999 = 4975,
so N = 5000 - 4975 = \boxed{25}.
```

Figure 2: The two output modes of METATHINK. The `<meta>` block resides inside `<think>` and contains the metacognitive reasoning assessment. In CoT mode, reasoning content follows `</meta>` before `</think>`; in non-CoT mode, `</think>` immediately follows `</meta>`.

yields substantially better accuracy-efficiency tradeoffs, confirming the advantage of richer decision representations.

3.2 MULTI-TOKEN METACOGNITIVE REASONING

Given a problem x, a reasoning policy π_θ generates a response y. Standard reasoning models use `<think>` *reasoning* `</think>` *answer*. METATHINK adds a multi-token `<meta>`...`</meta>` assessment at the start of the `<think>` region to guide the reasoning decision:

$$y = \texttt{<think>}\ \underbrace{\texttt{<meta>}\, m \,\texttt{</meta>}}_{\text{metacognitive reasoning}}\ \underbrace{r}_{\text{reasoning}}\ \texttt{</think>}\ \underbrace{a}_{\text{answer}}$$

where m is a brief natural language self-assessment, r is the CoT reasoning between `</meta>` and `</think>`, and a is the final answer. Skipping CoT leaves r empty, so `</think>` immediately follows `</meta>`. Adaptive reasoning maximizes the fraction of problems with $r = \emptyset$ while preserving accuracy.

The assessment m ends with an explicit decision, such as "I should think step-by-step" or "I should not think step-by-step," determining whether r is populated (Figure 2). Whereas AdaptThink (Zhang et al., 2025a) encodes this decision in one token after `<think>`, METATHINK evaluates problem characteristics before committing and provides an interpretable explanation.

3.3 ASYMMETRIC REWARD DESIGN

Our asymmetric reward encourages efficiency without external difficulty annotations: correct direct answers receive a higher reward than correct CoT responses, while incorrect responses receive zero task reward. This encourages the model to skip CoT when confident in answering directly:

$$R_{\text{task}}(x, y) = \begin{cases} \lambda & \text{if } \texttt{correct}(a, a^*) \wedge r = \emptyset \\ 1.0 & \text{if } \texttt{correct}(a, a^*) \wedge r \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where a^* is the ground-truth answer and $\lambda > 1$ controls the efficiency incentive. Difficulty is defined dynamically by the model's evolving capabilities: a problem is "easy" when the current model can solve it without detailed reasoning. Smaller λ values (e.g., 1.1) preserve higher accuracy with moderate CoT reduction; larger values encourage more direct answers, providing control over the accuracy–efficiency tradeoff.

A format consistency reward aligns the stated metacognitive decision with the model's actual behavior:

$$R_{\text{fmt}}(y) = \alpha \cdot \mathbb{1}\big[\texttt{fmt_ok}(y) \wedge \texttt{meta_con}(y)\big] \tag{2}$$

where $\alpha = 0.2$. The predicate $\texttt{fmt_ok}(y)$ requires exactly one </meta> and one </think> tag in the correct order, with non-empty assessment and answer. The predicate $\texttt{meta_con}(y)$ checks that r is empty or non-empty as declared. The total reward is $R(x, y) = R_{\text{task}}(x, y) + R_{\text{fmt}}(y)$.

To prevent the model from placing detailed reasoning in the answer to obtain the higher non-CoT reward, we set $R_{\text{task}}(x, y) \leftarrow 0$ whenever $r = \emptyset \wedge |y| > L_{\max}$. This length penalty encourages concise direct answers. Adaptive methods also risk decision boundary collapse into a single mode; AdaCoT (Lou et al., 2025) requires explicit countermeasures. METATHINK mitigates collapse by distributing the decision across a richer multi-token representation, as demonstrated in §4.3. Because difficulty follows the model's evolving capabilities, the reward also adapts across architectures and scales.

Training with GRPO. We train METATHINK using Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which estimates advantages by comparing G sampled responses per problem without a separate critic. The group-relative advantage is $\hat{A}_i = (R(x, y_i) - \mu_G)/(\sigma_G + \epsilon)$, and the policy is updated via a clipped surrogate objective with KL regularization against a reference policy. Prior to RL training, the model undergoes a lightweight supervised fine-tuning stage on a small set of examples to familiarize it with the metacognitive reasoning output format. The actual adaptive reasoning policy is learned entirely through RL. The complete procedure is summarized in Algorithm 1 (Appendix C).

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and cold start. We split DeepScaleR's approximately 40K math problems (Luo et al., 2025) into non-overlapping sets: 4K for cold-start SFT, 1K for the MathHard evaluation benchmark, and the remainder for RL. For R1-distilled models, we construct cold-start SFT targets entirely from model self-sampling: the model generates the meta assessment, thinking content, and final answer. We concatenate these components as generated to form each SFT target. This SFT-data construction requires no external annotations.

Models. We use DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025) for the main experiments. We also conduct a complete set of experiments on the non-thinking base models Qwen2.5-1.5B and Qwen2.5-7B (Qwen et al., 2025), including the main comparisons and additional analyses; see Appendix A.

Training Details. We use $\lambda \in \{1.1, 1.2, 1.3, 1.4, 1.5\}$, a non-CoT length threshold $L_{\max}$ of 640 tokens for 1.5B and 1024 for 7B, and a format reward coefficient $\alpha = 0.2$. All RL-trained methods share the same GRPO configuration: learning rate 10^{-6} without warmup, batch size 128, group size $G = 8$, two epochs, and low-variance KL loss with $\beta = 0.001$. We train with verl (Sheng et al., 2025) and the vLLM inference backend (Kwon et al., 2023) on 8× NVIDIA H100 80GB GPUs. All RL-trained methods receive cold-start SFT on the same 4K split with method-specific prompt templates, followed by their respective RL objectives. AdaCoT uses DeepSeek Chat for difficulty annotation; method-specific hyperparameters for AdaCoT and AdaptThink follow their original papers. Unless otherwise noted, results for trained models average three independent training runs with different random seeds.

Compared Methods. **Original Thinking/NoThinking** use the R1-distilled models' thinking/direct-answer modes. **Vanilla CoT/Non-CoT RL** train with standard GRPO on the same DeepScaleR data and always use CoT or answer directly, respectively. **AdaCoT** (Lou et al., 2025) learns CoT selection through a Pareto-optimal RL objective using external difficulty annotations. **AdaptThink** (Zhang et al., 2025a) uses annotation-free, constrained RL to skip CoT while maintaining accuracy. **AutoThink** (Tu et al., 2026) learns when to skip reasoning through multi-stage RL with progressive reward shaping.

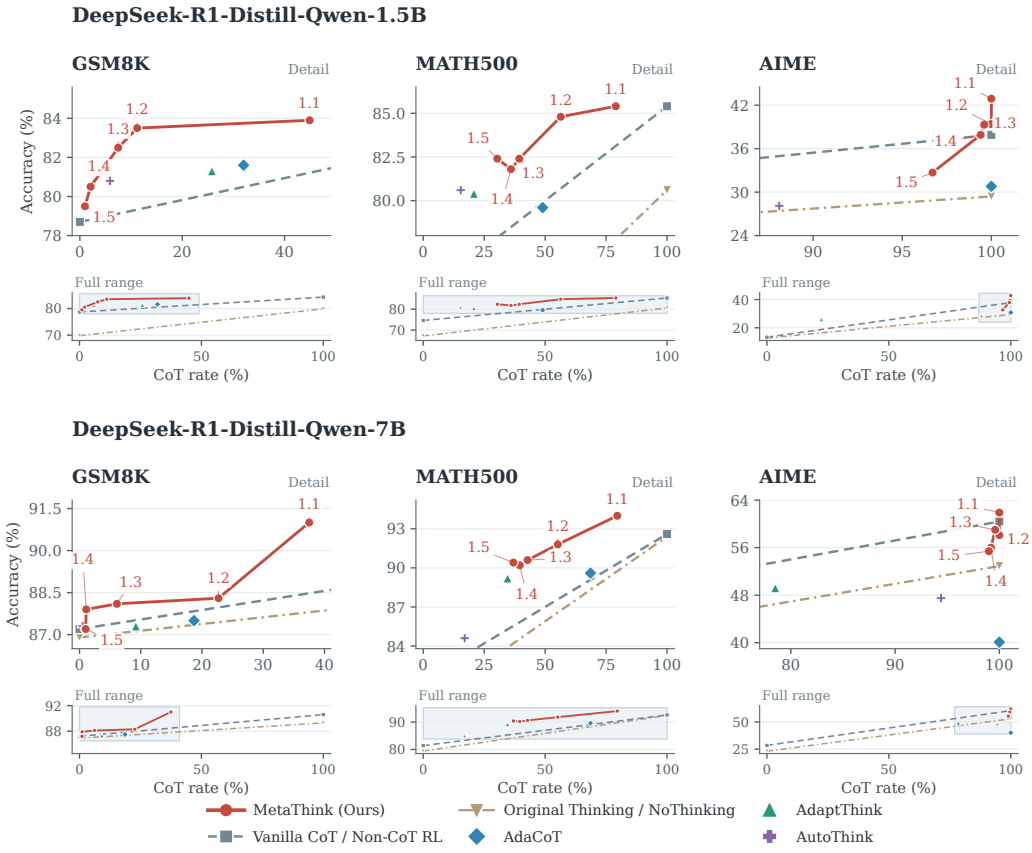


Figure 3: Accuracy–CoT tradeoffs on GSM8K, MATH500, and AIME 2024 for DeepSeek-R1-Distill-Qwen-1.5B (top) and 7B (bottom). Each panel pairs a detail view with a full-range overview; shading marks the enlarged region. Labels indicate λ, and METATHINK points are connected in increasing λ order. Dashed lines connect the CoT and non-CoT endpoints of each baseline family.

Evaluation. We evaluate both models on **GSM8K** (Cobbe et al., 2021), **MATH500** (Hendrycks et al., 2021; Lightman et al., 2024), and **AIME 2024** (MAA, 2024). Unless otherwise noted, all models use identical inference settings: `temperature=0.7`, `top_p=0.95`, `top_k=-1`, and `max_tokens=16384`. Benchmark comparisons average 16 repeated evaluations; intervention settings appear in Appendix E. We report accuracy (Acc, %), CoT usage (CoT Rate, %), and average generated tokens (Avg), counting assessments, reasoning, answers, and delimiters.

4.2 MAIN RESULTS

Figure 3 and Table 1 show that METATHINK achieves stronger accuracy–efficiency tradeoffs across both model scales. On GSM8K, the 7B model at $\lambda = 1.1$ reaches 91.0% accuracy with only 37.5% CoT usage, outperforming Vanilla CoT RL's 90.6% accuracy at 100.0% CoT. On MATH500, it likewise improves accuracy while reducing CoT usage, demonstrating effective metacognitive control.

METATHINK also surpasses existing adaptive policies. On GSM8K (1.5B), $\lambda = 1.3$ achieves 82.5% accuracy at a 7.5% CoT rate, outperforming AdaCoT and AdaptThink in both metrics. Similar gains on MATH500 and the 7B model show that varying λ provides effective control over the accuracy–efficiency tradeoff.

On AIME, METATHINK achieves the highest accuracy among all compared methods: 42.9% for 1.5B and 61.9% for 7B at $\lambda = 1.1$, exceeding Vanilla CoT RL by 5.0 and 1.5 percentage points. Its

Table 1: Main results on GSM8K, MATH500, and AIME 2024 for DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B. Acc and CoT denote accuracy and CoT rate, both in percent. Avg reports the average number of generated tokens.

Method	GSM8K			MATH500			AIME 2024		
	Acc	CoT	Avg	Acc	CoT	Avg	Acc	CoT	Avg
DeepSeek-R1-Distill-Qwen-1.5B									
Original Thinking	79.9	100.0	1017	80.6	100.0	4655	29.4	100.0	12372
Original NoThinking	69.8	0.0	281	67.2	0.0	945	12.7	0.0	5919
Vanilla CoT RL	84.3	100.0	1645	85.4	100.0	3752	37.9	100.0	10507
Vanilla Non-CoT RL	78.7	0.0	285	74.6	0.0	683	13.3	0.0	2057
AdaCoT	81.6	32.0	870	79.6	49.0	900	30.8	100.0	9275
AdaptThink	81.3	25.8	412	80.4	20.8	1650	26.0	22.3	11006
AutoThink	80.8	5.9	516	80.6	15.4	2410	28.1	88.1	9276
METATHINK ($\lambda = 1.1$)	83.9	44.9	884	85.4	79.0	2557	42.9	100.0	10722
METATHINK ($\lambda = 1.2$)	83.5	11.2	536	84.8	56.4	2693	39.4	100.0	10202
METATHINK ($\lambda = 1.3$)	82.5	7.5	419	82.4	39.4	2440	39.3	99.6	9851
METATHINK ($\lambda = 1.4$)	80.5	2.1	336	81.8	36.0	2300	37.9	99.4	9622
METATHINK ($\lambda = 1.5$)	79.5	1.0	281	82.4	30.4	2150	32.7	96.7	8481
DeepSeek-R1-Distill-Qwen-7B									
Original Thinking	89.3	100.0	703	92.4	100.0	3640	52.9	100.0	10470
Original NoThinking	86.9	0.0	283	79.4	0.0	848	23.1	0.0	3701
Vanilla CoT RL	90.6	100.0	944	92.6	100.0	2291	60.4	100.0	10236
Vanilla Non-CoT RL	87.2	0.0	228	81.4	0.0	576	28.5	0.0	1800
AdaCoT	87.5	18.7	658	89.6	68.6	3074	40.1	100.0	5291
AdaptThink	87.3	9.2	272	89.2	34.6	1923	49.2	78.5	8956
AutoThink	87.3	0.5	244	84.6	17.0	1328	47.5	94.4	8326
METATHINK ($\lambda = 1.1$)	91.0	37.5	547	94.0	79.6	1978	61.9	100.0	8666
METATHINK ($\lambda = 1.2$)	88.3	22.7	375	91.8	55.2	2061	58.1	100.0	7628
METATHINK ($\lambda = 1.3$)	88.1	6.1	310	90.6	42.8	1916	59.0	99.6	8223
METATHINK ($\lambda = 1.4$)	87.9	1.1	223	90.2	39.6	1799	56.0	99.2	8115
METATHINK ($\lambda = 1.5$)	87.2	1.0	242	90.4	37.0	1732	55.4	99.0	7610

reasoning allocation follows benchmark difficulty. At $\lambda = 1.3$, the 1.5B model uses CoT on 7.5% of GSM8K, 39.4% of MATH500, and 99.6% of AIME, preserving extensive reasoning for challenging problems.

These gains extend to total generation cost. For 7B at $\lambda = 1.1$, mean GSM8K output length falls from 944 to 547 tokens relative to Vanilla CoT RL. MATH500 and AIME also require fewer tokens while achieving higher accuracy, confirming improved performance with less generation.

4.3 SINGLE-TOKEN DECISION ABLATION

We compare METATHINK with **METATHINK-ST** using DeepSeek-R1-Distill-Qwen-1.5B at $\lambda = 1.1$. We construct a separate cold-start SFT dataset for METATHINK-ST from the METATHINK examples, keeping the sample count, input problems, CoT/non-CoT target proportions, and all content outside the meta block identical. Only the meta content changes: the model's original decision is encoded as `YES` when thinking is needed and `NO` otherwise. Both variants use the same number of SFT training steps, followed by RL with the same hyperparameters. The resulting SFT target formats are:

CoT: `<think><meta>YES</meta>` ***CoT*** `</think>` ***answer***
Non-CoT: `<think><meta>NO</meta></think>` ***answer***

Figure 4 shows that METATHINK-ST collapses to nearly zero CoT within 100 steps, with responses shortening to a few hundred tokens, while METATHINK stabilizes at 60–70% CoT usage and about 4,300 tokens. The multi-token stage thus sustains adaptive reasoning.

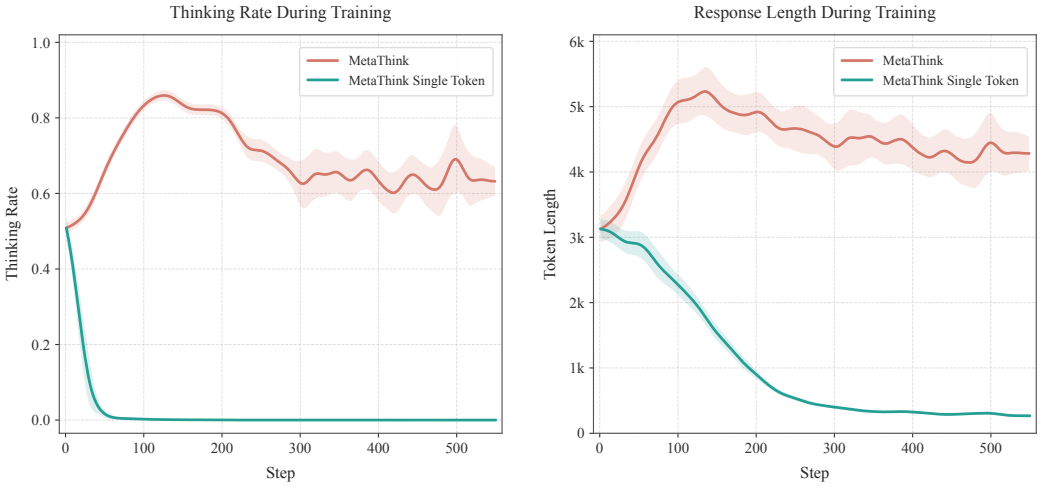


Figure 4: Single-token ablation on DeepSeek-R1-Distill-Qwen-1.5B at $\lambda = 1.1$: CoT rate (left) and response length (right) during RL. METATHINK-ST collapses to non-CoT generation, while METATHINK maintains both modes.

To separate learned content from extra autoregressive computation, we further add routing interventions, forced-CoT comparisons, and a planning-prompt control (Appendix E). Original meta improves routing discrimination and forced-CoT accuracy, while generic planning fails to reproduce the gains. These results support the effectiveness of learned, problem-specific metacognitive content.

4.4 META-BLOCK CONTENT AND LENGTH

On DeepSeek-R1-Distill-Qwen-1.5B at $\lambda = 1.1$, meta blocks average 28.8–35.2 tokens with 79.8–100.0% distinct texts (Table 2). Removing the fixed suffix ("So I should [not] think step-by-step.") after whitespace normalization leaves these rates unchanged on all four benchmarks. Thus, the brief meta blocks contain diverse assessments beyond the shared decision wording.

Table 2: Meta-block diversity and length (DeepSeek-R1-Distill-Qwen-1.5B, $\lambda = 1.1$). n counts evaluated responses. Distinct meta is $100U/n$, where U counts unique whitespace-normalized texts before or after removing the fixed decision suffix. Mean meta tokens is the average full meta-block length.

Benchmark	**Samples** (n)	**Distinct meta (%)**		**Mean meta tokens**
		Full text	Without suffix	
GSM8K	1,319	79.8	79.8	28.8
MATH500	500	97.4	97.4	31.0
AIME 2024	30	100.0	100.0	35.2
MathHard	1,000	98.5	98.5	33.4

Non-CoT meta audit. Among 1,751 non-CoT responses, no meta block contains two or more mathematical expressions (0 samples, 0.00%), and the mean meta length is 28.1 tokens. These results show that non-CoT meta blocks remain concise assessments of the problem's reasoning requirements.

Learned content beyond the decision sentence. Appendix D shows eight learned meta blocks across different reasoning demands (Figure 8). The assessments identify concrete operations, from transformations and vector analysis for CoT to direct conversion and recall for non-CoT. Each assessment precedes the shared decision sentence. Together with the single-token ablation in §4.3,

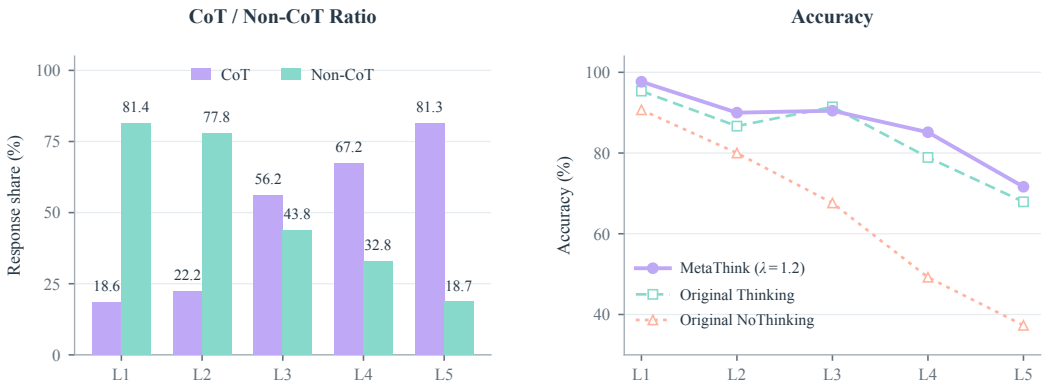


Figure 5: Difficulty-based reasoning on MATH500 with DeepSeek-R1-Distill-Qwen-1.5B. Left: CoT and non-CoT response shares for METATHINK at $\lambda = 1.2$. Right: accuracy compared with Original Thinking and Original NoThinking.

these results demonstrate that the multi-token stage learns problem-specific assessments rather than a canned binary gate.

4.5 NON-COT RESPONSE LENGTH

To examine whether non-CoT generation conceals long reasoning traces, we measure non-CoT rollout lengths on MATH500 with DeepSeek-R1-Distill-Qwen-1.5B (Table 3). Outputs remain compact across all five λ settings; at $\lambda = 1.1$, the mean and median are 385 and 335 tokens. Together with the meta-block audit (§4.4), this supports efficient direct answering rather than systematic relocation of long reasoning into the answer.

Table 3: Non-CoT rollout lengths on MATH500 for DeepSeek-R1-Distill-Qwen-1.5B, measured in tokens. Each column contains only the non-CoT rollouts at the corresponding λ.

λ	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**
Mean tokens	385	427	373	388	411
Median tokens	335	313	328	329	346

4.6 DIFFICULTY AND COT USAGE

Figure 5 shows that METATHINK adapts reasoning to problem difficulty. At $\lambda = 1.2$, the 1.5B model answers 81.4% of Level 1 problems without CoT while achieving 97.7% accuracy; CoT usage rises to 81.3% on Level 5. On Levels 4 and 5, it outperforms both Original Thinking and Original NoThinking, demonstrating stronger accuracy through selective reasoning. Appendix B presents all five λ settings.

5 CONCLUSION

We presented METATHINK, which introduces a multi-token metacognitive reasoning stage and an asymmetric reward for adaptive CoT reasoning without external annotations. Experiments with R1-distilled models across two model scales and three benchmarks show that METATHINK achieves higher accuracy than full-CoT baselines while reducing CoT usage, outperforming existing adaptive reasoning methods. The single-token ablation and meta-block analyses further show that the multi-token stage learns diverse, problem-specific assessments and sustains adaptive reasoning. Additional Qwen2.5 experiments demonstrate its effectiveness when training from non-thinking base models.

Ethics Statement

Our method reduces computational cost by skipping chain-of-thought reasoning on problems the model deems easy. A potential risk is that the model may occasionally misjudge problem difficulty and skip reasoning on problems that require it, leading to incorrect answers in safety-critical applications. We recommend that practitioners validate the accuracy-efficiency tradeoff on their target domain before deployment.

AI Use Statement

Large language models only assisted with language revision, manuscript organization and visualization. Large language models were not involved in designing or conducting the experiments.

Reproducibility Statement

To facilitate reproducibility, Section 3 describes the MetaThink framework, including the multi-token metacognitive reasoning format, asymmetric reward, format consistency reward, and non-CoT length constraint. Section 4.1 specifies the dataset splits, cold-start data construction, model backbones, training hyperparameters, computational resources, compared methods, and evaluation protocols. Appendix C provides pseudocode for the training algorithm. Section 4.3 and Appendix A.3 describe the controlled single-token ablations, while Appendix E details the meta-content interventions and their evaluation settings. Appendix A reports additional experiments with Qwen2.5 base models, including output-length statistics and routing analyses. Appendix B provides difficulty-level results across reward settings, and Appendix D presents representative learned meta blocks.

References

Rakefet Ackerman and Valerie A Thompson. Meta-reasoning: Monitoring and control of thinking and reasoning. *Trends in cognitive sciences*, 21(8):607–617, 2017.

Anthropic. Introducing claude opus 4.6. https://www.anthropic.com/news/claude-opus-4-6, 2026.

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

Jonathan St BT Evans and Keith E Stanovich. Dual-process theories of higher cognition: Advancing the debate. *Perspectives on psychological science*, 8(3):223–241, 2013.

Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. Efficient reasoning models: A survey. *arXiv preprint arXiv:2504.10903*, 2025.

Google DeepMind. Gemini 3 pro: the frontier of vision ai. https://blog.google/innovation-and-ai/technology/developers-tools/gemini-3-pro-vision/, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu,

Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, Sept 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware llm reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24842–24855, 2025.

Michael Hassid, Gabriel Synnaeve, Yossi Adi, and Roy Schwartz. Don't overthink it. preferring shorter thinking chains for improved llm reasoning. *arXiv preprint arXiv:2505.17813*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Advances in Neural Information Processing Systems*, 38:162239–162262, 2026.

Daniel Kahneman. *Thinking, fast and slow*. macmillan, 2011.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. Adacot: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning. *arXiv preprint arXiv:2505.11896*, 2025.

Jinghui Lu, Haiyang Yu, Siliang Xu, Shiwei Ran, Guozhi Tang, Siqi Wang, Bin Shan, Teng Fu, Hao Feng, Jingqun Tang, et al. Prolonged reasoning is not all you need: Certainty-based adaptive routing for efficient llm/mllm reasoning. *arXiv preprint arXiv:2505.15154*, 2025.

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. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://github.com/agentica-project/deepscaler, 2025. Notion Blog.

Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. Reasoning models can be effective without thinking. *arXiv preprint arXiv:2504.09858*, 2025.

MAA. American invitational mathematics examination (AIME) 2024. Mathematical Association of America, 2024. URL https://maa.org/maa-invitational-competitions/.

Sara Vera Marjanović, Arkil Patel, Vaibhav Adlakha, Milad Aghajohari, Parishad BehnamGhader, Mehar Bhatia, Aditi Khandelwal, Austin Kraft, Benno Krojer, Xing Han Lù, et al. Deepseek-r1 thoughtology: Let's think about llm reasoning. *arXiv preprint arXiv:2504.07128*, 2025.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

OpenAI. Learning to reason with llms. *OpenAI Blog*, 2024.

OpenAI. Introducing gpt-5.4. https://openai.com/index/introducing-gpt-5-4/, 2026.

Qianjun Pan, Wenkai Ji, Yuyang Ding, Junsong Li, Shilian Chen, Junyi Wang, Jie Zhou, Qin Chen, Min Zhang, Yulan Wu, et al. A survey of slow thinking-based reasoning llms using reinforced learning and inference-time scaling law. *arXiv preprint arXiv:2505.02665*, 2025.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025. URL https://arxiv.org/abs/2412.15115.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 10131–10165, 2025.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, 2025.

Valerie A Thompson, Jamie A Prowse Turner, and Gordon Pennycook. Intuition, reason, and metacognition. *Cognitive psychology*, 63(3):107–140, 2011.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. Learning when to think: Shaping adaptive reasoning in r1-style models via multi-stage rl. *Advances in Neural Information Processing Systems*, 38:15181–15207, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *arXiv preprint arXiv:2203.11171*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Hao Wen, Xinrui Wu, Yi Sun, Feifei Zhang, Liye Chen, Jie Wang, Yunxin Liu, Yunhao Liu, Ya-Qin Zhang, and Yuanchun Li. Budgetthinker: Empowering budget-aware llm reasoning with control tokens. *arXiv preprint arXiv:2508.17196*, 2025.

Zishun Yu, Tengyu Xu, Di Jin, Karthik Abinav Sankararaman, Yun He, Wenxuan Zhou, Zhouhao Zeng, Eryk Helenowski, Chen Zhu, Sinong Wang, et al. Think smarter not harder: Adaptive reasoning with inference aware optimization. *arXiv preprint arXiv:2501.17974*, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3716–3730, 2025a.

Xiaoyun Zhang, Jingqing Ruan, Xing Ma, Yawen Zhu, Haodong Zhao, Hao Li, Jiansong Chen, Ke Zeng, and Xunliang Cai. When to continue thinking: Adaptive thinking mode switching for efficient reasoning. *arXiv preprint arXiv:2505.15400*, 2025b.

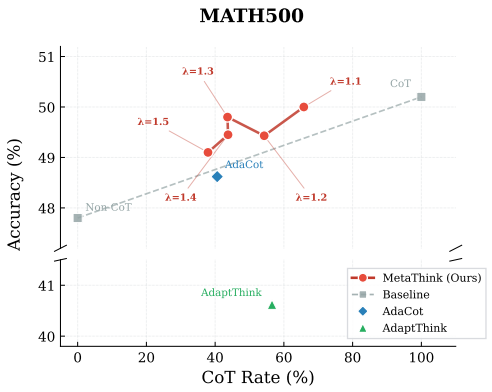
Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. Marco-o1: Towards open reasoning models for open-ended solutions. *arXiv preprint arXiv:2411.14405*, 2024.

A TRAINING FROM QWEN2.5 BASE MODELS

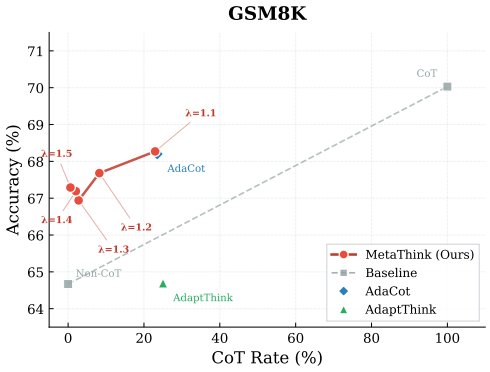
Building on the DeepSeek-R1-Distill experiments in the main paper, we further evaluate METATHINK on models that do not natively support a dedicated thinking mode. We apply the same cold-start and reinforcement-learning recipe to Qwen2.5-1.5B and Qwen2.5-7B (Qwen et al., 2025), first teaching the metacognitive output format and then learning when to use CoT through the asymmetric reward. The results extend the effectiveness of METATHINK to non-thinking initializations at both model scales. We first present the Pareto curves and main results, followed by further analyses of generation cost, decision representation, response behavior, and reasoning allocation.

A.1 MAIN RESULTS

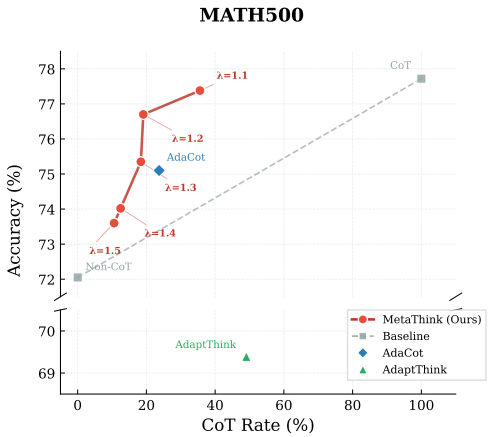
Figure 6 and Table 4 present the accuracy–efficiency tradeoffs obtained from Qwen2.5-1.5B and Qwen2.5-7B.



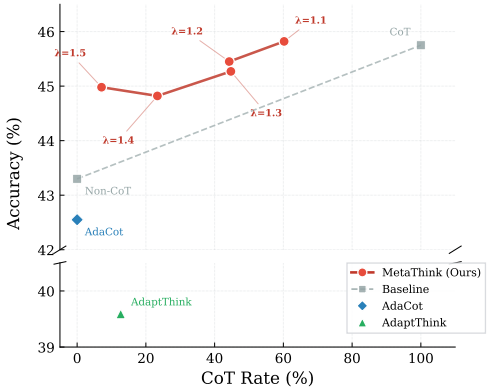
(a) 1.5B, MATH500



(b) 1.5B, GSM8K



(c) 7B, MATH500



(d) 7B, MathHard

Figure 6: Accuracy–CoT tradeoffs for Qwen2.5-1.5B (top) and Qwen2.5-7B (bottom). METATHINK points correspond to $\lambda \in \{1.1, 1.2, 1.3, 1.4, 1.5\}$. Dashed lines connect the full-CoT and non-CoT RL baselines.

Table 4: Main results for the Qwen2.5 base models. Qwen2.5-1.5B is evaluated on MATH500 and GSM8K; Qwen2.5-7B on MATH500 and MathHard. All values are percentages.

Qwen2.5-1.5B				
Method	**MATH500**		**GSM8K**	
	Acc (%)	CoT Rate (%)	Acc (%)	CoT Rate (%)
Baseline CoT RL	50.2	100.0	70.0	100.0
Baseline Non-CoT RL	47.8	0.0	64.7	0.0
AdaCoT	48.6	40.6	68.2	23.5
AdaptThink	40.6	56.6	64.7	25.0
METATHINK (λ=1.1)	50.0	65.8	68.3	23.0
METATHINK (λ=1.2)	49.4	54.3	67.7	8.2
METATHINK (λ=1.3)	49.8	43.6	66.9	2.8
METATHINK (λ=1.4)	49.5	43.8	67.2	2.1
METATHINK (λ=1.5)	49.1	37.9	67.3	0.6
Qwen2.5-7B				
Method	**MATH500**		**MathHard**	
	Acc (%)	CoT Rate (%)	Acc (%)	CoT Rate (%)
Baseline CoT RL	77.7	100.0	45.8	100.0
Baseline Non-CoT RL	72.1	0.0	43.3	0.0
AdaCoT	75.1	23.7	42.6	0.0
AdaptThink	69.4	49.1	39.6	12.7
METATHINK (λ=1.1)	77.4	35.6	45.8	60.2
METATHINK (λ=1.2)	76.7	19.1	45.5	44.3
METATHINK (λ=1.3)	75.4	18.4	45.3	44.7
METATHINK (λ=1.4)	74.0	12.5	44.8	23.4
METATHINK (λ=1.5)	73.6	10.6	45.0	7.1

We evaluate Qwen2.5-1.5B on MATH500 and GSM8K, and Qwen2.5-7B on MATH500 and the held-out MathHard split. The experiments follow the cold-start and RL training recipe in §4.1, with $L_{\max} = 640$ for 1.5B and $L_{\max} = 1024$ for 7B. We compare against Baseline CoT RL, Baseline Non-CoT RL, AdaCoT, and AdaptThink, and vary λ from 1.1 to 1.5.

METATHINK achieves strong accuracy–efficiency tradeoffs at both model scales. On MATH500, the 1.5B model at $\lambda = 1.1$ reaches 50.0% accuracy with 65.8% CoT usage, close to Baseline CoT RL's 50.2% accuracy at 100.0% CoT. For 7B, the same setting reaches 77.4% accuracy with only 35.6% CoT usage, compared with the full-CoT baseline's 77.7%. On MathHard, it achieves 45.8% accuracy with a 60.2% CoT rate, matching the full-CoT baseline while substantially reducing reasoning use.

The method also improves on existing adaptive policies. On MATH500 (7B), $\lambda = 1.2$ reaches 76.7% accuracy at a 19.1% CoT rate, surpassing AdaCoT's 75.1% accuracy at 23.7% CoT and AdaptThink's 69.4% at 49.1% CoT. At $\lambda = 1.5$, the 7B model reaches 73.6% accuracy on MATH500 with 10.6% CoT usage, outperforming the non-CoT baseline's 72.1%. The 1.5B model reaches 67.3% accuracy on GSM8K with just 0.6% CoT usage, outperforming its non-CoT baseline's 64.7%. These results show that METATHINK learns effective adaptive reasoning directly from non-thinking base models, and that λ provides control over the accuracy–efficiency tradeoff.

The following Qwen2.5-7B analyses provide further evidence for effective metacognitive control from a non-thinking initialization. We use MATH500 and MathHard, with GSM8K and AIME additionally included in the ablation and meta-block analyses. Metacognitive assessments typically contain 20–50 tokens.

A.2 TOTAL GENERATION COST

To further assess generation efficiency from a non-thinking initialization, we measure complete output costs, including the metacognitive assessment, CoT, answer, and generated delimiters. Table 5 reports average output tokens for Qwen2.5-7B; the full mean and median statistics appear in Section A.5. Specifically, the metacognitive stage adds only 28.6 tokens on average (approximately 29), including the generated `</meta>` delimiter. This overhead is included in the total output counts.

Table 5: Mean total output tokens per response (Qwen2.5-7B), including metacognitive overhead.

Method	**MATH500**	**MathHard**
Baseline CoT RL	2302.6	4044.6
Baseline Non-CoT RL	438.0	565.8
AdaCoT	1371.6	1716.8
AdaptThink	468.8	758.8
METATHINK (λ=1.1)	1515.3	3091.1
METATHINK (λ=1.2)	1280.7	2623.5
METATHINK (λ=1.3)	1068.2	2104.3
METATHINK (λ=1.4)	969.8	1976.7
METATHINK (λ=1.5)	750.8	1470.4

Relative to Baseline CoT RL, $\lambda = 1.1$ reduces mean output length by 34.2% on MATH500 and 23.6% on MathHard. At $\lambda = 1.5$, the reductions reach 67.4% and 63.6%, respectively. These reductions extend the generation savings observed in the main experiments to Qwen2.5-7B, including the metacognitive overhead.

A.3 SINGLE-TOKEN DECISION ABLATION

Building on the R1-distilled ablation in §4.3, we repeat the comparison between METATHINK and **METATHINK-ST** on Qwen2.5-7B to further verify the role of multi-token decisions. We construct a separate cold-start SFT set with the same sample count, input problems, CoT/non-CoT target proportions, and training steps as METATHINK. Only the meta content is replaced with `YES` or `NO` according to the model's original thinking decision; all other target content remains identical. Both variants then use the same RL hyperparameters.

Table 6: CoT rates (%) at the ablation endpoints on Qwen2.5-7B.

Method	**Train**	**Validation**		
		MATH500	MathHard	AIME 2024
METATHINK (λ=1.1)	58	35.6	60.2	95
METATHINK-ST (λ=1.1)	2	0	1	0
METATHINK-ST (λ=1.5)	1	0	1	2

Single-token decisions collapse; multi-token decisions remain adaptive. For both λ settings, the training CoT rate falls below 5% within the first ten RL steps and ends at 2% (λ=1.1) or 1% (λ=1.5). Its validation CoT rates are only 0–2% (Table 6). In contrast, the multi-token policy at λ=1.1 retains both modes and allocates CoT more often to harder benchmarks: 35.6% on MATH500, 60.2% on MathHard, and 95% on AIME.

Reduced CoT use comes with accuracy losses. At λ=1.1, replacing the multi-token assessment with the binary switch reduces accuracy from 77.4% to 69.4% on MATH500, from 45.8% to 37.0%

on MathHard, and from 16.7% to 8.5% on AIME. These correspond to losses of 8.0, 8.8, and 8.2 percentage points; the AIME loss is about 49% relative. The drop on GSM8K is smaller, from 91.1% to 88.9%. The switch therefore loses accuracy alongside its selective use of CoT, including on the hardest benchmark.

Together with the R1-distilled ablation, these results further show that multi-token metacognitive reasoning prevents decision collapse, preserves difficulty-dependent CoT allocation, and enables stable, effective adaptive reasoning across thinking and non-thinking initializations.

A.4 Response and Meta-Block Audits

Non-CoT response length audit. To extend the content and length analysis in §4.4 to a non-thinking initialization, we further examine non-CoT response lengths on Qwen2.5-7B by comparing them with Baseline Non-CoT RL (Table 8). For METATHINK, median lengths on MATH500 range from 368 to 399 tokens, close to the baseline's 365.5 tokens. On MathHard, the medians range from 480 to 560.5 tokens, compared with 508.0 for the baseline. These results show that typical non-CoT responses remain concise and comparable in length to direct-answer generation, complementing the total token savings in §A.2.

Meta-block diversity. The Qwen2.5-7B results also show that the meta stage learns varied assessments across problems. Among 500 MATH500 outputs, 481 whitespace-normalized meta strings are distinct (96.2%); among 1,000 MathHard outputs, 958 are distinct (95.8%). These counts remain unchanged after removing the fixed decision sentence, and the most frequent assessment accounts for only 1.0% of each benchmark. Thus, the diversity extends beyond the final binary decision phrase: the model generates a range of assessments before selecting a reasoning mode. Together with the single-token ablations, these results provide further support for the multi-token metacognitive design across model initializations.

A.5 Additional Output-Length Statistics

To support these additional efficiency analyses, we report the complete mean and median output-token statistics for Qwen2.5-7B, supplementing the total-cost analysis and non-CoT response audit in §A.2 and §A.4. Total output counts include the metacognitive assessment, reasoning, answer, and generated delimiters.

Table 7: Average (Avg) and median (Med) output token counts per sample (Qwen2.5-7B).

Method	**MATH500**		**MathHard**	
	Avg	Med	Avg	Med
Baseline CoT RL	2302.6	1180.0	4044.6	3107.5
Baseline Non-CoT RL	438.0	365.5	565.8	508.0
AdaCoT	1371.6	761.0	1716.8	1223.0
AdaptThink	468.8	285.5	758.8	487.0
METATHINK (λ=1.1)	1515.3	491.0	3091.1	2031.0
METATHINK (λ=1.2)	1280.7	466.0	2623.5	1219.5
METATHINK (λ=1.3)	1068.2	447.0	2104.3	980.0
METATHINK (λ=1.4)	969.8	403.0	1976.7	695.0
METATHINK (λ=1.5)	750.8	420.0	1470.4	646.0

Non-CoT lengths. Table 8 reports the average and median output lengths of non-CoT responses. Across reward settings, the median lengths remain close to those of Baseline Non-CoT RL.

A.6 Decision Quality

To extend the reasoning-allocation analysis in §4.6 to a non-thinking initialization, we examine how λ controls Qwen2.5-7B's decisions using two categories defined by Baseline Non-CoT RL and Baseline CoT RL. Category A contains problems answered incorrectly by Non-CoT RL but correctly by CoT

Table 8: Non-CoT response length statistics (Qwen2.5-7B). Avg and Med are the average and median token counts of non-CoT responses.

Method	MATH500		MathHard	
	Avg	Med	Avg	Med
Baseline Non-CoT RL	438.0	365.5	565.8	508.0
AdaCoT	1074.4	575.0	1716.8	1223.0
AdaptThink	382.7	234.0	537.9	372.0
METATHINK (λ=1.1)	525.1	369.0	896.4	480.0
METATHINK (λ=1.2)	560.4	399.0	861.5	560.5
METATHINK (λ=1.3)	482.0	392.0	704.4	517.0
METATHINK (λ=1.4)	496.1	368.0	700.7	497.0
METATHINK (λ=1.5)	464.2	392.0	565.1	525.5

RL. Category B contains problems both baselines answer correctly. We measure **Necessary Rate**, the proportion of Category A on which the adaptive model uses CoT, and **Efficiency Rate**, the proportion of Category B on which it skips CoT.

Table 9: Routing on MATH500 categories defined by baseline outcomes (Qwen2.5-7B). Necessary is CoT usage in Category A; Efficiency is non-CoT usage in Category B.

Method	Necessary (%)	Efficiency (%)
AdaCoT	23.50	75.98
AdaptThink	84.25	42.00
METATHINK (λ=1.1)	60.00	88.22
METATHINK (λ=1.2)	34.00	94.80
METATHINK (λ=1.3)	36.00	95.95
METATHINK (λ=1.4)	15.25	97.83
METATHINK (λ=1.5)	12.00	98.55

Table 9 presents the results. As λ increases from 1.1 to 1.5, the Efficiency Rate rises steadily from 88.22% to 98.55%, while the Necessary Rate decreases from 60.00% to 12.00%. This confirms that λ acts as a meaningful control knob for the accuracy-efficiency tradeoff. A higher λ more aggressively encourages direct answering, causing the model to skip CoT on a larger fraction of problems, including some that benefit from reasoning.

These results extend the main reasoning-allocation findings to Qwen2.5-7B: larger λ values favor direct answers in Category B, while smaller values retain more CoT in Category A. Thus, METATHINK also learns to balance efficiency and reasoning from a non-thinking initialization.

B Difficulty-Level Analysis Across Reward Settings

Figure 7 extends the $\lambda = 1.2$ analysis in §4.6 to all five reward settings.

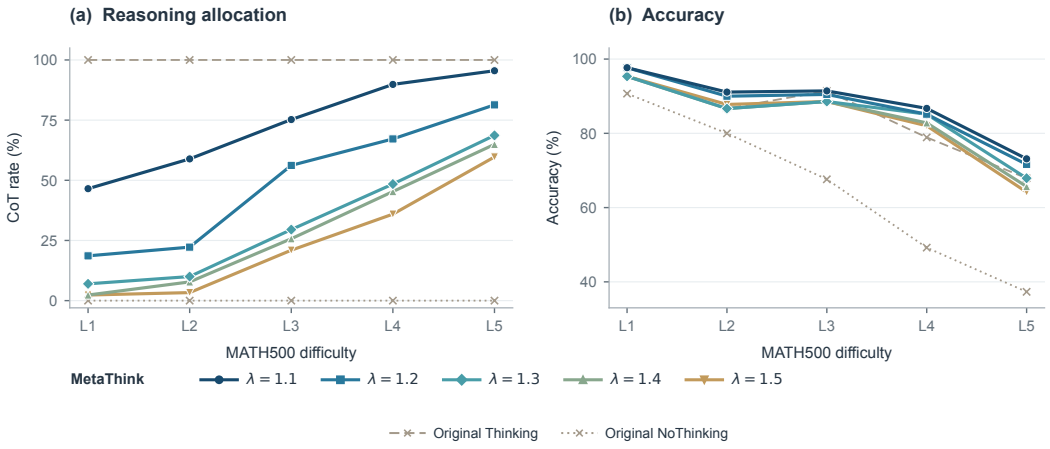


Figure 7: Complete difficulty-level results on MATH500 with DeepSeek-R1-Distill-Qwen-1.5B. Left: CoT usage for all five λ settings and the Original baselines. Right: accuracy at each difficulty level. Original Thinking always uses CoT, while Original NoThinking always answers directly.

Across all settings, CoT usage increases with problem difficulty, while larger λ values encourage more direct answers. On Level 1, increasing λ from 1.1 to 1.5 reduces CoT usage from 46.5% to 2.3%, with accuracy remaining at 95.3% or higher. On Level 5, the corresponding CoT rates are 95.5% and 59.7%, showing that the policy continues to devote substantial reasoning to the hardest problems. Lower-λ variants achieve the strongest accuracy on these problems: at $\lambda = 1.1$, Level 4 and Level 5 accuracies reach 86.7% and 73.1%, compared with 78.9% and 67.9% for Original Thinking. Together, these results show that λ provides effective control over reasoning effort while preserving a consistent allocation of more CoT to harder problems. MATH500 difficulty labels are used only for evaluation, not for training or routing.

C TRAINING ALGORITHM

Algorithm 1 METATHINK Training

Require: Policy model π_θ; dataset $\mathcal{D}$; hyperparameters $\lambda, \alpha, G, \beta, L_{\max}$

Initialize reference policy $\pi_{\text{ref}} \leftarrow \pi_\theta$
for step $= 1, \ldots, M$ **do**
  Sample mini-batch $\mathcal{D}_b \subset \mathcal{D}$
  for each problem $x \in \mathcal{D}_b$ **do**
    Sample G responses $\{y_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid x)$
    for $i = 1, \ldots, G$ **do**
      Parse $y_i = (m_i,\ r_i,\ a_i)$
      Compute $R_{\text{task}}(x, y_i)$ via Eq. 1
      Compute $R_{\text{fmt}}(y_i)$ via Eq. 2
      if $r_i = \emptyset \wedge |y_i| > L_{\max}$ **then**
        $R_{\text{task}}(x, y_i) \leftarrow 0$
      end if
      $R(x, y_i) \leftarrow R_{\text{task}}(x, y_i) + R_{\text{fmt}}(y_i)$
    end for
    $\hat{A}_i \leftarrow (R(x, y_i) - \mu_G)/(\sigma_G + \epsilon)$
  end for
  Update π_θ via GRPO with clipped surrogate + KL
end for
return π_θ

D Representative Learned Meta Blocks

We show the full learned meta blocks discussed in §4.4, generated by DeepSeek-R1-Distill-Qwen-1.5B at $\lambda = 1.1$. The examples span algebra, geometry, arithmetic, and graph interpretation, illustrating how the assessments identify the reasoning demands of each problem before selecting CoT or non-CoT generation.

CoT selected

Double summation

It requires multi-step reasoning involving transformations, substitutions, and summation techniques to express the given double sum in terms of known constants p and q.

So I should think step-by-step.

Perfect cubes

It requires multi-step reasoning involving algebraic manipulation and checking of perfect cubes to find the smallest solution.

So I should think step-by-step.

Vector geometry

It requires multi-step vector analysis to compute the angle between two lines defined by specific scalar multiples and then find the angle using the dot product formula.

So I should think step-by-step.

Order of operations

It requires multi-step reasoning involving the correct placement of parentheses to achieve different results based on the order of operations and the grouping of terms.

So I should think step-by-step.

Non-CoT selected

Coordinate conversion

It is a standard conversion from rectangular to polar coordinates with a clear, direct method and a single answer.

So I should not think step-by-step.

Linear substitution

It is a straightforward substitution of values into a linear function and basic arithmetic, which is a simple and linear process.

So I should not think step-by-step.

Divisor counting

It is a standard mathematical question requiring only a direct recall of divisor counting techniques.

So I should not think step-by-step.

Graph interpretation

It is a standard question requiring only basic interpretation of a graph to determine which point has the highest average speed.

So I should not think step-by-step.

Figure 8: Representative learned meta blocks from DeepSeek-R1-Distill-Qwen-1.5B at $\lambda = 1.1$. Each card shows the full meta text; headings identify the problem type. The assessment appears in black and the final decision sentence in color, separating the learned assessment from the shared decision wording.

E INTERVENING ON METACOGNITIVE CONTENT

We test three interventions on MATH500 using DeepSeek-R1-Distill-Qwen-1.5B: METATHINK at $\lambda = 1.2$ and Vanilla CoT RL. Category A contains problems solved by CoT RL but not Non-CoT RL; Category B contains those solved by both. Generation uses 10 samples per problem with temperature 1.0 and top-p 1.0.

Controls. We replace the original meta with length-matched text from another problem (same or opposite decision) or generic neutral text. Routing AUROC separates Categories A and B using the log-probability difference between think and do-not-think continuations; we report both summed and per-token-normalized scores. We also force CoT to isolate the effect on reasoning, and add a 30-token generic planning instruction to Vanilla CoT RL.

Table 10: Meta-content interventions. Panel (a): routing AUROC and the share of replacements within two tokens of the original meta length. Panels (b,c): accuracy (%) and mean generated tokens in Category A.

(a) Routing and meta-length matching

Meta	AUROC (sum)	AUROC (mean/token)	Within 2 tokens (%)
Original	0.686	0.685	–
Shuffled (same)	0.662	0.661	99.8
Shuffled (opposite)	0.268	0.270	86.6
Neutral	0.476	0.457	93.8

(b) Forced CoT: effect on reasoning

Meta body	Acc, Category A	Acc, Category B
Original	88.33	98.56
Shuffled (same)	78.33	96.54
Neutral	86.00	97.41

(c) Vanilla CoT RL with generic planning

Variant	Acc, Category A	Acc, Category B	Tokens, Category A
Plain always-CoT	83.33	95.39	5,886
+ 30-token planning instruction	70.00	92.22	6,237

Results. Original meta gives the strongest routing discrimination under both scoring conventions. With CoT forced for every problem, it improves Category A accuracy by 10.0 points over shuffled meta and 2.33 over neutral text. Generic planning instead reduces Vanilla CoT RL accuracy by 13.3 points in Category A and 3.17 in Category B while increasing token use. Together, these controls support the value of learned, problem-specific metacognitive content beyond extra tokens or generic planning.