

Concise Math Reasoning via Difficulty-Aware Distillation

Anonymous ACL submission

Abstract

Human experts tackle difficult math problems by identifying and executing a few pivotal steps rather than listing every intermediate thought. In contrast, standard Chain-of-Thought (CoT) distillation trains small models on lengthy reasoning traces, encouraging a uniform overthinking style across easy and hard items alike. The result is rigid, slow solutions that sacrifice adaptivity. This approach stands in sharp contrast to human intuition. Humans naturally adapt their problem-solving strategy, dedicating significant effort to difficult problems while finding quick, simple solutions for easier ones. We argue that the root cause lies in the training data: it contains excess information and reasoning steps organized in ways misaligned with human practice. We address this with Difficulty-Aware Distillation (DAD), a procedure for producing training data that mirrors concise human reasoning. A large teacher model first assesses a problem’s difficulty and then rewrites the solution to retain only the essential steps. Using this process, we constructed LiteCoT, a 100,000-example corpus of short, clear rationales, and used it to train our *Liter* models. With 100k LiteCoT, we outperform models trained on 800k long CoT and cut both training and inference costs. The advantage is consistent across standard math benchmarks, showing that concise, human-aligned data delivers equal or better accuracy with much less compute. For example, on the challenging AIME24 exam, our approach reaches 74.2% Pass@1 using only about 5K inference tokens, surpassing other methods that consume many more tokens.

1 Introduction

To improve the mathematical reasoning capabilities of large language models (LLMs), a common approach is to distill “Chain-of-Thought” (CoT) from powerful teacher models like DeepSeek-R1 (Guo et al., 2025) and OpenAI-o1 (OpenAI, 2025). However, this approach suffers from a fundamental flaw:

the training data it relies on is inherently overly lengthy (Ye et al., 2025; Concise). These CoTs are filled with repetitive analytical steps, forcing the student model to learn a rigid pattern of “overthinking” in all mathematical problems (Yu et al., 2025a; Sui et al., 2025; Yin et al., 2025). This mechanical process is fundamentally misaligned with how humans approach mathematical problem solving: humans typically construct reasoning paths by focusing on a few key steps, without needing to record every intermediate thought. Notably, previous efforts aimed at improving data quality, such as data filtering (Muennighoff et al., 2025; Wen et al., 2025), synthesis (Yang et al., 2025) or selecting for difficulty (Ye et al., 2025) have largely ignored this core issue of verbosity. This neglect results in high training costs for current models and an inability to flexibly adjust their reasoning depth based on the actual difficulty of the mathematical problem.

Perhaps the key to improving the mathematical reasoning of LLMs is to have them reason more like humans—by focusing only on essential logical steps. This would require training data where solutions are concise and their length is adapted to the problem’s difficulty. To explore this hypothesis, we introduce a simple method called Difficulty-Aware Distillation (DAD). DAD is designed to create this type of data: it uses a teacher model to assess a math problem’s difficulty and then rewrites the lengthy original solution into a compact trace containing only the essential logical steps.

Using the DAD method, we built **LiteCoT**, a dataset of 100K concise samples averaging just 720 tokens—a stark contrast to the 5K-10K tokens in datasets like S1 (Muennighoff et al., 2025) and OpenThoughts (Team, 2025b). We used this data to train our *Liter* models. The results are compelling: across 9 benchmarks, models trained on our 100K LiteCoT samples consistently outperform those trained on 800K verbose examples. On the challenging AIME24 benchmark, *Liter* achieves 74.2%

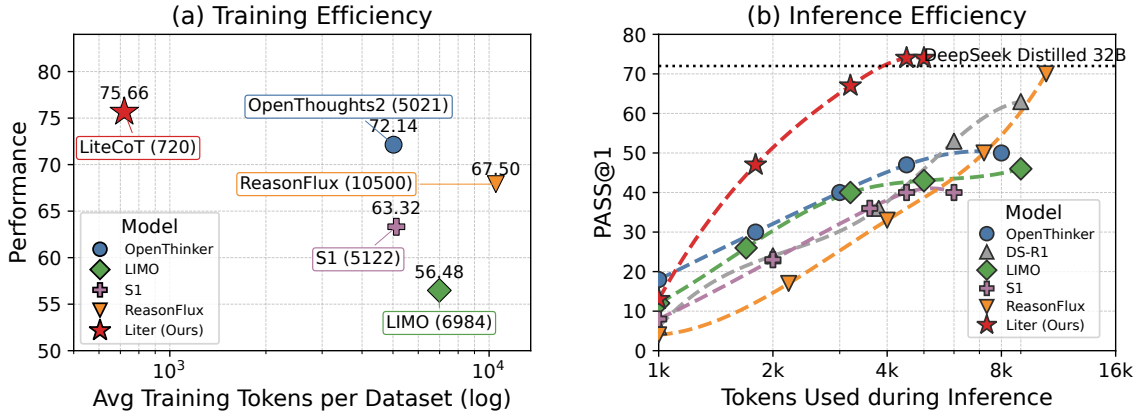


Figure 1: (a) Training tokens comparison of LiteCoT with prior distillation methods (32B model); (b) Inference efficiency and accuracy on the challenging AIME24 benchmark (32B model). Our LITER models trained with LiteCoT achieve higher accuracy (74.2% Pass@1) at substantially reduced inference token counts compared to prior methods.

Pass@1 with superior token efficiency, proving that for math reasoning, data quality and conciseness trump sheer quantity.

2 Related Work

2.1 LLM Distillation for Math Reasoning

Recent studies have shown that distilling the long Chain-of-Thought (CoT) reasoning from large teacher models to smaller student models is an effective method for enhancing their mathematical reasoning abilities (Guo et al., 2025; Wei et al., 2022). However, the training and inference costs of these methods remain high, with the root cause being that the CoT reasoning itself is overly verbose and consumes substantial computational resources. To alleviate this cost pressure, current mainstream research has focused on the data level, optimizing the training set by, for example, filtering for high-quality samples (Ye et al., 2025; Team, 2025b,a) or rewriting and synthesizing data (Yang et al., 2025; Guan et al., 2025). Nevertheless, these methods only address the issue of how much data to use, while overlooking a more central point: the selected reasoning chains themselves are still replete with unnecessary reasoning steps (Qu et al., 2025; Wu et al., 2025; Luo et al., 2025b). Little work has reflected on whether such lengthy reasoning chains are truly necessary for logically rigorous mathematical reasoning. We argue that the core of mathematical reasoning lies in logical clarity and procedural precision, not in the sheer volume of the process. Overly long reasoning chains not only introduce significant computational overhead, but their redundant information can even become

“noise” that interferes with the model’s learning performance (Lin et al., 2025; Hou et al.; Zhao et al., 2025b).

2.2 Reasoning Pruning via Budget Control

Beyond the aforementioned approaches that focus on data quantity, more advanced work has begun to directly address the cost issues caused by CoT length itself. However, most of these strategies concentrate on inference-time optimizations (Muennighoff et al., 2025; Teng et al., 2025), such as LightThinker (Zhang et al., 2025), Sketch-of-Thought (Aytes et al., 2025), and AdaR1 (Luo et al., 2025a), but they do not solve the efficiency problem during the training phase. In contrast, the exploration of training efficiency remains relatively underexplored. Current methods for this purpose fall into two main categories: (1) Learning-based approaches (Yu et al., 2025c; Zhao et al., 2025a) that train auxiliary models to produce shorter reasoning traces, which can adaptively switch between Long and Short CoTs but often incur substantial additional training costs. (2) Prompt engineering-based pipelines that heuristically rewrite CoTs without training extra models, such as DLCoT (Luo et al., 2025b) and Long-Short Chain-of-Thought Mixture (Yu et al., 2025b).

3 LiteCoT Dataset: Concise Reasoning with Big Gains

3.1 Difficulty-Aware Prompting for Curating LiteCoT

We now describe our difficulty-aware distillation (DAD) pipeline, designed to construct a concise

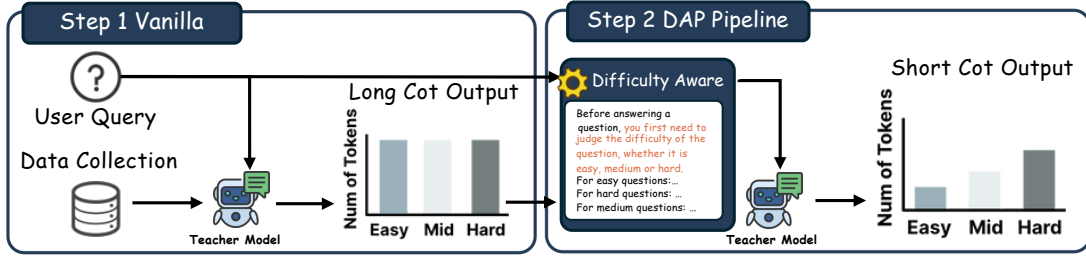


Figure 2: Overview of the DAD method. First, the DeepSeek R1 model generates initial Long CoT data from collected user queries. Then, a difficulty-aware prompt guides the model to rewrite concise CoT outputs, adapting reasoning length dynamically based on problem difficulty.

and difficulty-adaptive distilled reasoning dataset. The pipeline consists of two sequential steps: (1) generating initial long reasoning solutions, and (2) refining these chains using difficulty-aware prompts to produce shorter, complexity-aligned reasoning traces.

Step 1 (Long CoT Generation). In conventional distillation, a teacher LLM $M_{teacher}$ is used to generate a reasoning chain $CoT_{L,i}$ for each query Q_i in the training set D . Formally: $CoT_{L,i} = M_{teacher}(Q_i)$ for $i = 1, 2, \dots, N$. The collected set of long chains is $C_L = (Q_i, CoT_{L,i})_{i=1}^N$. These Long CoTs often contain redundant content and do not adjust in length for query difficulty.

Step 2 (Difficulty-aware CoT Refinement). In this step, the teacher model takes each $(Q_i, CoT_{L,i})$ pair as input, along with a specially designed prompt P_{DA} that instructs it to produce a shorter solution appropriate for the difficulty of Q_i . This prompt P_{DA} provides different guidelines depending on whether a question is easy, medium, or hard. For example, it might include: “For easy questions: provide a brief solution. . . For hard questions: include all necessary detailed steps . . .”. Guided by P_{DA} , the teacher first assesses the difficulty of the problem and then refines the solution accordingly:

1. **Difficulty Assessment:** Guided by P_{DA} , the model first assesses the difficulty of the input query Q_i , classifying it into a predefined set of difficulty levels $\mathcal{K} = \{k_{easy}, k_{medium}, k_{hard}\}$. This assessment process can be represented as:

$$d_i = M_{teacher}(Q_i, CoT_{L,i}, P_{DA}) \quad (1)$$

where $d_i \in \mathcal{K}$ is the assessed difficulty level. The prompt P_{DA} contains explicit instructions for different assessments.

2. **Difficulty-Adaptive CoT Refinement:** Based

on the assessed difficulty level d_i , the model follows the sub-prompt or structured output template $P_{d_i} \subset P_{DA}$ within P_{DA} designed for that specific difficulty d_i . It then prunes, reorganizes, or guides the generation of a new, difficulty-matched short CoT, CoT_{S,d_i} . This process can be represented as:

$$CoT_{S,d_i} = M_{teacher}((Q_i, CoT_{L,i}), P_{DA}, d_i) \quad (2)$$

Or more specifically, if P_{d_i} is an explicit output template:

$$CoT_{S,d_i} = M_{teacher}(\text{input} = (Q_i, CoT_{L,i}), \text{template} = P_{d_i}(d_i)) \quad (3)$$

where $P_{d_i}(d_i)$ denotes the specific instruction part within P_{DA} for difficulty d_i (e.g., “For easy questions: . . .”, “For medium questions: . . .”, “For hard questions: . . .”).

Applying this process to all training samples yields a new distilled dataset $C_S = (Q_i, CoT_{S,i})_{i=1}^N$ comprised of shortened, difficulty-graded reasoning traces.

In addition, the refined CoTs are much shorter than the originals (for easy and medium queries, $Length(CoT_{S,i}) \ll Length(CoT_{L,i})$). This adaptive reduction strategy is expected to significantly improve the efficiency of model training while preserving essential reasoning for challenging tasks. Ultimately, a student model trained on C_S should learn to solve problems with an appropriate amount of reasoning.

3.2 LiteCoT Overview

LiteCoT Overview. Using the DAD method, we created a distilled dataset, named LiteCoT, of 100K question–solution pairs with difficulty-adaptive CoTs. Table 1 highlights the token savings of our approach compared to prior distillation

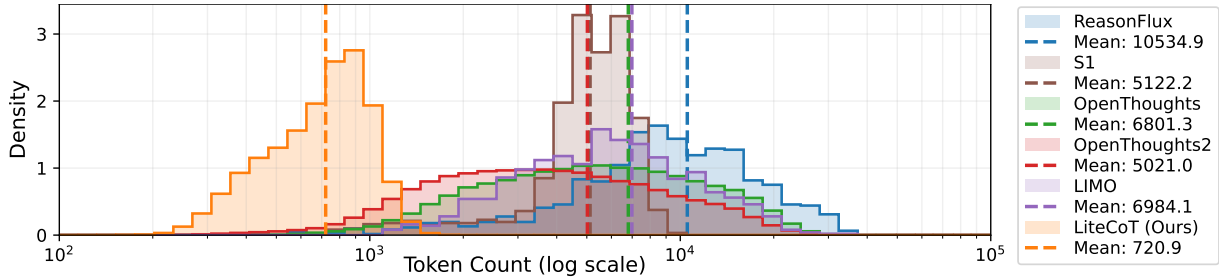


Figure 3: Token count distribution of LiteCoT compared with previous CoT distillation datasets.

Table 1: Performance comparison of different budget control methods. “ Δ Tokens” denotes average tokens per CoT sample.

Datasets	#-Samples	Δ Tokens
Light-R1 (Wen et al., 2025)	75.9K	6382
OpenThoughts (Team, 2025b)	113.9K	6801
OpenThoughts2 (Team, 2025b)	1M	5021
S1 (Muennighoff et al., 2025)	1K	5122
LIMO (Ye et al., 2025)	817	6984
SKY-T1 (Team, 2025a)	16K	4095
ReasonFlux (Yang et al., 2025)	1K	10500
LiteCoT (ours)	100K	720

data. Other budget-control methods still produce CoT solutions averaging several thousand tokens each (e.g., 5K–10K tokens for ReasonFlux (Yang et al., 2025) and OpenThoughts (Team, 2025b)), often necessitating tens or hundreds of thousands of training samples. In stark contrast, LiteCoT’s CoT samples average only about 720 tokens, representing an order-of-magnitude improvement in token efficiency.

Figure 3 clearly shows this advantage: the token-count distribution for LiteCoT is markedly shifted toward shorter reasoning chains compared to other datasets.

Training the LITER with LiteCoT. Using LiteCoT, we distilled a new family of reasoning models, named LITER (1.5B, 7B, and 32B), based on the Qwen2.5. We followed the training strategy from Open-R1 (Face, 2025) and fine-tuned the LITER models for 3 epochs on the LiteCoT dataset. Specifically, we employed a learning rate of $5e-5$ with a cosine learning rate scheduler, including a 10% warm-up phase. In addition, we increased the RoPE frequency to 100K, effectively extending the context length from 4K to 8K tokens.

ected to yield substantial benefits in terms of reduced training duration, lower computational resource demands, and potentially more focused learning for student models, the empirical valida-

tion of which will be discussed in the subsequent experimental sections.

4 Experiments

We conduct extensive experiments to evaluate the effectiveness of LiteCoT data. Our evaluations address three key questions: **(Exp-1)** Does training on short difficulty-aware CoTs, LiteCoT, actually improve model performance compared to long CoTs? **(Exp-2)** Can our DAD method rival or surpass models distilled with an order of magnitude more data from a powerful teacher? **(Exp-3)** How does our DAD approach compare to other recent CoT distillation and compression techniques? **(Exp-4)** Will DAD still be useful if we change another teacher model? **(Exp-5)** How many tokens can be saved using DAD method?

Due to space limitations, we have placed the details of Exp 4 and 5 in the appendix A.

4.1 Experimental Results and Analysis

4.1.1 Exp-1: LiteCoT vs. Long CoT Baselines

We evaluate the fine-tuned models trained on our LiteCoT and Long CoT data, comparing their overall performance and analyzing the accuracy-speed trade-off.

Overall Performance. As shown in Table 2, we comprehensively compare the average accuracy of Qwen2.5 series models and Llama3.1-8B trained with Short CoT and Long CoT on 11 different benchmarks. The average performances show that all models trained and distilled with short CoT data significantly outperform the corresponding models distilled with Long CoT data. For example, the overall average performance of Qwen2.5-1.5B-Instruct (short) reached 43.6%, while its Long CoT version was 35.7%; Qwen2.5-7B-Instruct (short) reached 57.3%, higher than the 53.3% of the Long CoT version. This trend is consistently reflected in models of different parameter sizes.

Table 2: Accuracy of Long CoT distilled models and short CoT distilled models. Green means short CoT distilled models outperform the corresponding Long CoT distilled models. Pink means the opposite (“QW” and “LM” refer to Qwen2.5-Instruct and Llama3.1-Instruct).

Benchmark	QW-0.5B		QW-1.5B		QW-3B		QW-7B		QW-14B		LM-8B	
	short	long	short	long	short	long	short	long	short	long	short	long
Math Reasoning												
MATH500	18.6	11.6	62.8	56.4	47.8	46.0	66.2	64.8	65.0	64.4	34.2	32.2
AIME24	0.0	0.0	3.3	0.0	3.3	0.0	23.3	3.3	13.3	10.0	3.3	0.0
GSM8K	33.6	20.5	85.1	84.4	77.3	61.4	95.5	95.0	90.8	85.1	73.7	26.8
Minerva Math	3.7	4.4	25.4	19.5	14.3	11.8	33.9	31.2	29.0	21.3	10.3	7.7
GaoKao2023	23.4	11.9	65.2	62.9	43.3	39.5	66.5	66.2	56.9	49.9	32.2	25.2
OlympiadBench	5.9	1.5	37.2	36.3	15.4	10.8	42.8	37.5	28.4	19.1	13.0	11.6
College Math	19.6	9.8	46.9	47.2	35.6	31.2	48.5	47.5	41.9	32.1	24.7	1.61
Sat Math	37.5	12.5	43.8	3.1	68.8	34.4	96.9	90.6	96.9	62.5	43.8	25.0
ACM23	2.5	0.0	47.5	45.0	30.0	17.5	62.5	57.5	42.5	27.5	5.0	25.0
Cross Domain Reasoning												
MMLU STEM	29.7	9.2	33.9	10.4	58.1	21.4	64.4	60.5	71.3	56.9	33.9	29.0
GPQA	22.7	21.2	28.8	27.3	23.7	22.7	29.8	32.3	40.4	38.4	32.3	25.3
Overall	17.9	9.3	43.6	35.7	38.0	27.0	57.3	53.3	52.4	42.5	27.9	19.0

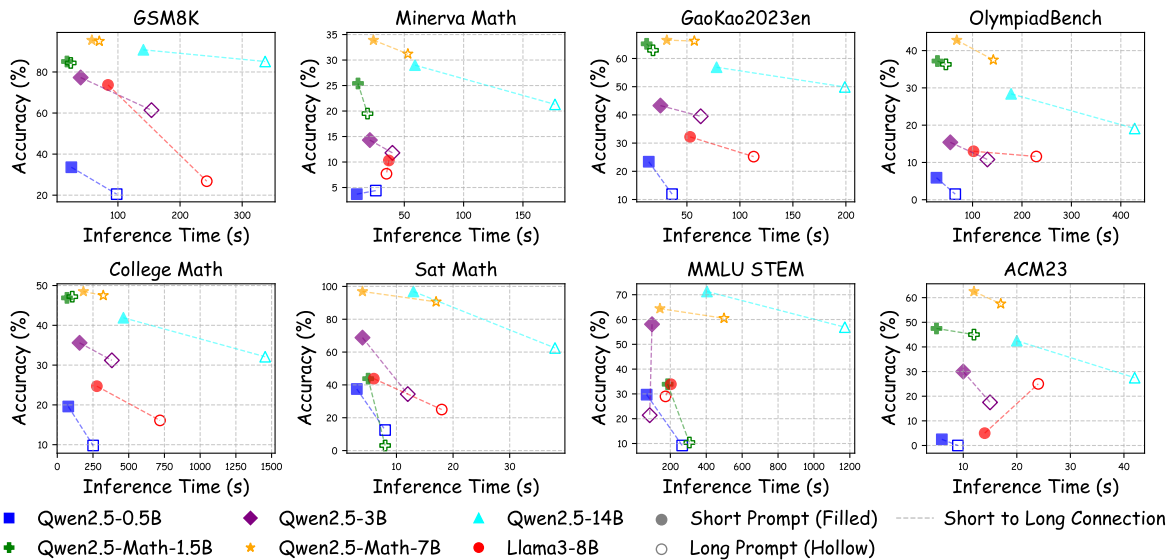


Figure 4: Inference time vs. accuracy across eight benchmarks. Models trained with short CoT (filled symbols) demonstrate significantly faster inference times compared to Long CoT (hollow symbols) while maintaining or exceeding accuracy levels.

The relationship between inference speed and accuracy. Figure 4 clearly shows that for all models (Llama3.1-8B and Qwen2.5 series models), the use of Short CoT (filled points) generally achieves inference time that is much lower than Long CoT (hollow points). More importantly, this significant speed improvement does not come at the expense of accuracy; on the contrary, in most benchmarks, Short CoT is not only faster, but its accuracy can also reach or even exceed the level of Long CoT. For example, on multiple benchmarks such as GSM8K and MMLU STEM, Short CoT data points all show a better “speed-accuracy” balance. This finding spans models of different parameter sizes

and a variety of task types, and strongly proves that the DAD Short CoT method can significantly improve inference efficiency while maintaining or even optimizing the model’s inference accuracy.

4.1.2 Exp-2: Comparing our LITER Models with Mainstream Distilled LLMs

We first compared our LITER models against models distilled from DeepSeek-R1 (Exp-2.1). To further evaluate the effectiveness of our LITER models, we compared our LITER 7B and 32B variants against a broad range of mainstream distilled models across multiple reasoning benchmarks (Exp-2.2).

Exp-2.1: Comparison with DeepSeek-R1-

Table 3: Performance Comparison of Language Models on Reasoning Benchmarks. Since there are too many models to be tested, we only select some of them for retesting, and we directly copy the results of all other models from their papers or blogs. † represents the copied results, ‡ represents the results retested by ourself. DS-R1-Distill represents the official Distilled Models in the DeepSeek-R1 report. OB represents the Olympiad Bench. **Bold** means the first performance and underline means the second performance.

Model	AIME24	MATH500	AMC23	Olympiad Bench	GPQA	OVERALL
Frontier Models						
DeepSeek-V3†	39.2	<u>90.2</u>	85.0	55.4	<u>59.1</u>	65.8
Mathstral-7B-V0.1†	0.0	57.8	37.5	21.5	-	-
DS-Coder-V2-Instruct†	13.3	75.3	57.5	37.6	-	-
Qwen2.5-Math-72B-Instruct†	30.0	85.6	70.0	<u>49.0</u>	-	-
NuminaMath-72B-CoT†	3.3	64.0	70.0	<u>32.6</u>	-	-
LLaMA3.1-405B-Instruct†	-	73.8	-	34.8	51.1	-
rStar-Math†	<u>43.3</u>	88.2	<u>80.0</u>	63.1	-	-
1.5B						
Qwen2.5-Math-1.5B-instruct†	10.0	60.0	60.0	38.1	-	-
ReasonFlux-1.5B†	20.0	70.4	72.5	49.0	-	-
DS-R1-Distilled-1.5B‡	<u>28.9</u>	<u>83.9</u>	55.0	31.4	<u>33.8</u>	46.6
Liter-1.5B‡	36.7	87.2	<u>60.0</u>	<u>42.7</u>	35.4	52.4
7B						
Qwen2.5-Math-7B-instruct†	13.3	82.6	62.6	39.0	-	-
Openthinker2-7B†	50.0	88.4	89.5	-	<u>49.3</u>	-
Bespoke-Stratos-7B†	20.0	82.0	-	-	37.8	-
ReasonFlux-7B†	36.7	88.6	80.0	54.8	-	-
DS-R1-Distilled-7B‡	55.5	92.8	77.5	49.8	49.1	64.9
Liter-7B‡	<u>53.3</u>	93.8	82.5	50.2	53.3	66.6
32B						
Qwen2.5-32B-instruct†	13.3	79.4	37.5	-	33.8	-
Openthinker2-32B‡	63.3	96.0	80.0	<u>56.3</u>	65.1	72.1
Light-R1-32B†	70.0	94.2	77.5	-	61.1	-
LIMO-32B‡	40.0	95.0	52.5	43.9	51.0	56.5
S1.1-32B‡	43.3	89.2	75.0	56.6	52.5	63.3
ReasonFlux-32B‡	73.3	<u>96.4</u>	70.0	34.7	63.1	67.5
Bespoke-Stratos-32B†	63.3	93.0	-	-	58.1	-
Sky-T1-32B-preview†	43.3	82.4	-	-	56.8	-
QwQ-32B-Preview‡	76.7	92.2	72.5	54.1	65.1	72.1
Qwen3-32B‡	81.4	91.6	80.0	53.9	65.6	74.5
DS-R1-Distilled-32B‡	<u>72.6</u>	95.2	90.0	53.0	62.1	<u>74.6</u>
Liter-32B‡	76.7	96.6	<u>87.5</u>	53.9	<u>63.6</u>	75.7

Distilled Models. Figure 5 shows the performance of four models of different parameter sizes (1.5B, 7B, 14B and 32B) in the Qwen2.5 series on six key benchmarks after three training methods: original weights (Vanilla), DeepSeek-R1 distilled models (DeepSeek-R1-Distilled) and our Liter. It shows that the performance of the model representing Liter (Gray blue area) is consistent and significantly better than the other two methods at all parameter scales. Specifically, taking the Qwen2.5-7B model as an example, our method achieved an accuracy of 60.6% on Math500, while DeepSeek-R1 distilled and the original model achieved 55.8% and 42.6% respectively; on the more challenging Olympiad-

Bench, our model scored 30.2%, also ahead of DeepSeek R1 distilled’s 21.1% and the original model’s 10.0%. This performance advantage is particularly evident in the Qwen2.5-32B model with a larger number of parameters. For example, on the AIME24 benchmark, the accuracy of the DAD distilled model is as high as 76.7%, far exceeding the 48.0% of DeepSeek R1 distillation and the 13.3% of the original model. These specific numerical comparisons strongly prove that the model trained on the LiteCoT can achieve comprehensive performance superiority over other training methods on multiple core benchmarks

Exp-2.2: Comparison with Mainstream

RLMs of Similar Parameter Sizes. We conducted an extensive horizontal comparison involving our LITER models (1.5B, 7B, and 32B) against various publicly available mainstream reasoning models of comparable parameter scales. Evaluations were carried out on five representative benchmarks: AIME24, MATH500, GPQA, AMC23, and OlympiadBench, with results presented in Table 3.

Overall, whether at the parameter scale of 1.5B, 7B, or 32B, the models trained by the DAD distillation method show comparable or better performance than the current mainstream or even top reasoning models. At the 1.5B level, Ours-1.5B achieved an accuracy of 87.2% on MATH500, significantly better than most of the same-level comparison models, and showed strong competitiveness on OlympiadBench (42.7%). Entering the 7B level, Ours-7B performs particularly well, achieving an accuracy of 93.8% on MATH500, and leading the other 7B parameter models listed in the table with 53.3% on AIME24 and GPQA respectively, while also being at the top level on OlympiadBench (50.2%). When the model scale is expanded to 32B, the performance advantage of Ours-32B is further consolidated, for example, it ranks among the top on multiple difficult benchmarks such as MATH500 (96.6%), AIME24 (76.7%), and GPQA (63.6%).

4.1.3 Exp-3: Comparison with other CoT Optimization methods.

We continue to compare models trained with DAD distilled data against those using other CoT decomposition techniques, focusing on overall performance and the speed-accuracy trade-off to analyze their key differences and advantages.

Overall Performance of Different CoT Decomposition Methods. Table 4 clearly shows the performance comparison of different CoT decomposition methods on multiple benchmarks, including the original baseline model (Vanilla), Chain-of-Draft (COD), two LLMingua-2 with different compression rates (Lingua-2(0.9) and Lingua-2(0.5)), BudgetAware(2048), and our DAD method (Ours). We explicitly treat COD as a static prompting baseline to isolate the advantages of DAD’s adaptive reasoning. Ours consistently outperforms COD, especially on complex benchmarks where static constraints fail. For instance, on Sat Math, Ours achieves 96.9% accuracy, drastically surpassing COD’s 50.0%. Similarly, Ours leads COD by 17.5% on ACM23 and 10.0% on AIME24. While

methods like LLMingua-2(0.9) or BudgetAware show competitive single-point results, DAD maintains superior stability and accuracy across all tasks. This gap demonstrates that difficulty-aware modulation effectively preserves critical logical steps that rigid, one-size-fits-all strategies like COD often truncate.

The relationship between inference speed and accuracy. Figure 6 fully demonstrates the superiority of the DAD distillation method (labeled as “Ours”, purple star with gold rim). In the tests across eight different benchmarks, the “Ours” method has the best trade-off in almost all speed-accuracy two-dimensional planes. Specifically, compared with other mainstream CoT decomposition techniques such as Chain-of-Draft, LLMingua-2, and BudgetAware, our method not only achieves the highest accuracy in most cases (for example, 62.5% on ACM23 and 96.9% on Sat Math), but also achieves the shortest or very competitive inference times. In comparison, other methods either compromise on accuracy (such as LLMingua-2 (0.5) or COD with high compression ratios) or far exceed “Ours” in reasoning time (such as LLMingua-2 (0.9) with low compression ratios on some tasks). This better “price/performance ratio” consistently achieved on multiple benchmarks strongly demonstrates the unique advantages of the DAD distillation method.

5 Conclusion

In this paper, we introduced the Difficulty-Aware Distillation method, a new pipeline for CoT distillation. Leveraging DAP, we created LiteCoT, a difficulty-adaptive CoT dataset consisting of 100K distilled reasoning examples with an average length of only 720 tokens per sample, far shorter than existing alternatives. Utilizing LiteCoT, we distilled and trained a new family of reasoning models named LITER (1.5B, 7B, and 32B), based on the Qwen2.5 architecture. Extensive experiments across diverse reasoning benchmarks clearly demonstrate the effectiveness of the LiteCoT dataset and LITER models, consistently outperforming previous approaches that rely on substantially larger datasets with uniformly verbose reasoning chains.

6 Limitations

While our approach demonstrates promising results in generating concise mathematical reasoning, we

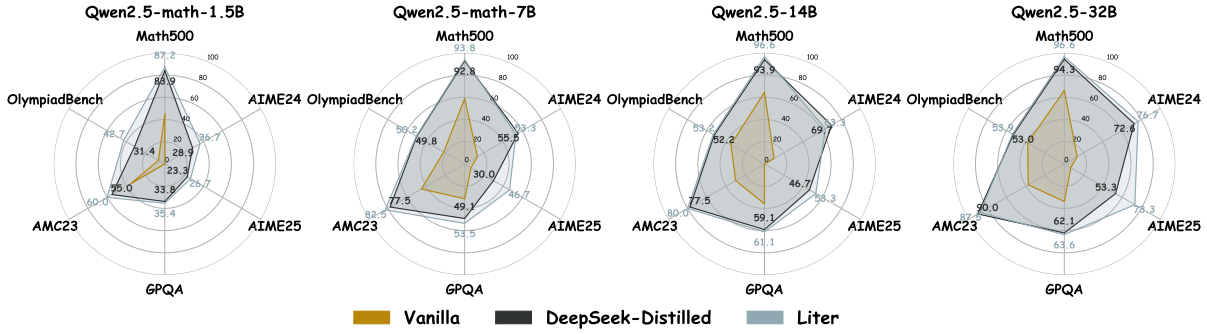


Figure 5: Radar chart show the performance comparison of the Vanilla, DeepSeek-R1-Distilled, and Liter versions of the Qwen2.5 model with different parameter sizes (1.5B, 7B, 14B, 32B) on multiple math and reasoning benchmarks (Math500, AIME24, OlympiadBench, AIME25, GPQA, AMC23).

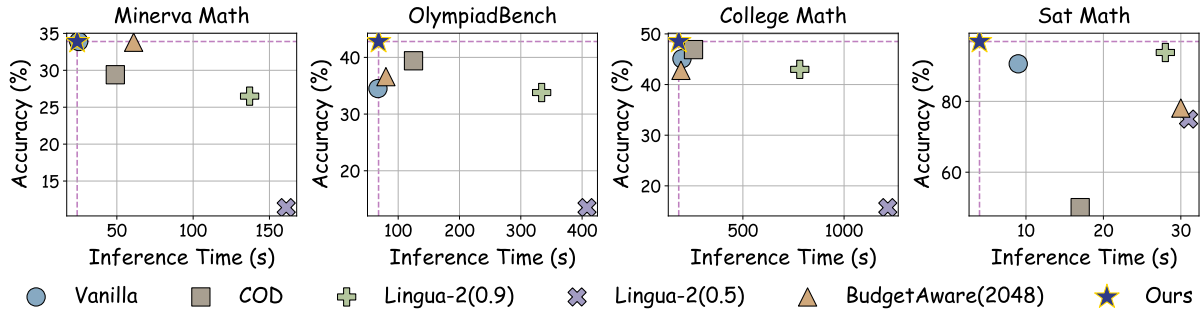


Figure 6: Scatter plots comparing inference time and accuracy of different methods across multiple benchmarks. Our method (Ours) consistently achieves optimal accuracy while significantly reducing inference time compared to existing methods.

Table 4: Performance comparison of different CoT decompose methods. Static Prompting represents Chain-of-Draft prompting method, Lingua-2 means LLMingua-2.

Benchmark	Vanilla	Static Prompting	Lingua-2(0.9)	Lingua-2(0.5)	BudgetAware(2048)	Ours
AIME24	13.3	13.3	16.7	0.0	13.3	23.3
AIME25	10.0	13.3	23.3	0.0	3.3	16.7
GSM8K	94.8	94.4	94.8	72.4	94.4	95.5
Minerva Math	33.9	29.4	26.5	11.4	33.8	33.9
GaoKao2023en	64.2	62.3	64.4	34.5	65.2	66.5
OlympiadBench	34.5	39.4	33.8	13.5	36.6	42.8
College Math	45.1	46.9	43.0	15.7	42.8	48.5
Sat Math	90.6	50.0	93.8	75.0	78.1	96.9
ACM23	60.0	45.0	45.0	27.5	52.5	62.5

443 acknowledge several limitations that offer avenues
444 for future research.

445 **Scope Limited to Text Modality** First, our
446 method’s effectiveness has been validated on text-
447 based tasks. Its applicability to multimodal reason-
448 ing, which involves diagrams or charts, remains
449 unexplored. This is primarily because generat-
450 ing high-quality, concise CoT data for multimodal
451 tasks is significantly more challenging, as it re-
452 quires grounding logical steps in visual informa-
453 tion.

454 **Suboptimal Performance in Code Generation**

455 Second, our method is less effective for code gener-
456 ation. Unlike mathematical reasoning, where
457 logical steps can often be condensed, code gener-
458 ation requires strict, step-by-step precision. Pruning
459 even a minor detail in a code’s logic can result in
460 syntax errors or a non-functional program, making
461 this approach less suitable for such tasks.

462 **Dependency on the Teacher Model** Finally, our
463 approach is heavily dependent on the capabilities
464 of the “teacher” model. The quality of the distilled
465 dataset is fundamentally capped by the teacher’s
466 ability to accurately assess difficulty and generate
467 logical, concise reasoning.

468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521

References

AI-MO. 2024. [Aime 2024](#).

AI-MO. 2025. [Aime 2025](#).

Simon A. Aytes, Jinheon Baek, and Sung Ju Hwang. 2025. Sketch-of-thought: Efficient llm reasoning with adaptive cognitive-inspired sketching. *arXiv preprint arXiv:2503.05179*.

Christopher A Choquette-Choo, Krishnamurthy Dvijotham, Krishna Pillutla, Arun Ganesh, Thomas Steinke, and Abhradeep Thakurta. 2023. Correlated noise provably beats independent noise for differentially private learning. *arXiv preprint arXiv:2310.06771*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

LLM Concise. Walk before you run! concise llm reasoning via reinforcement learning.

Hugging Face. 2025. [Open r1: A fully open reproduction of deepseek-r1](#).

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2024. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*.

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

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, 2025. URL <https://arxiv.org/abs/2504.01296>.

Weizhe Lin, Xing Li, Zhiyuan Yang, Xiaojin Fu, Hui-Ling Zhen, Yaoyuan Wang, Xianzhi Yu, Wu-long Liu, Xiaosong Li, and Mingxuan Yuan. 2025. Trimr: Verifier-based training-free thinking compression for efficient test-time scaling. *arXiv preprint arXiv:2505.17155*.

Haotian Luo, Haiying He, Yibo Wang, Jinluan Yang, Rui Liu, Naiqiang Tan, Xiaochun Cao, Dacheng Tao, and Li Shen. 2025a. [Adar1: From long-cot to hybrid-cot via bi-level adaptive reasoning optimization](#). *Preprint*, arXiv:2504.21659. 522
523
524
525
526

Yijia Luo, Yulin Song, Xingyao Zhang, Jiaheng Liu, Weixun Wang, GengRu Chen, Wenbo Su, and Bo Zheng. 2025b. Deconstructing long chain-of-thought: A structured reasoning optimization framework for long cot distillation. *arXiv preprint arXiv:2503.16385*. 527
528
529
530
531
532

math ai. 2025. [Minervamath](#). 533

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*. 534
535
536
537
538

OpenAI. 2025. [Introducing openai o1](#). 539

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, and 1 others. 2024. LlmLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *arXiv preprint arXiv:2403.12968*. 540
541
542
543
544
545

Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, and 1 others. 2025. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint arXiv:2503.21614*. 546
547
548
549
550
551

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*. 552
553
554
555
556

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, and 1 others. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*. 557
558
559
560
561
562

NovaSky Team. 2025a. Sky-t1: Train your own o1 preview model within \$450. <https://novasky-ai.github.io/posts/sky-t1>. Accessed: 2025-01-09. 563
564
565

Open Thoughts Team. 2025b. Open Thoughts. 566

Fengwei Teng, Zhaoyang Yu, Quan Shi, Jiayi Zhang, Chenglin Wu, and Yuyu Luo. 2025. Atom of thoughts for markov llm test-time scaling. *arXiv preprint arXiv:2502.12018*. 567
568
569
570

TIGER-Lab. 2024. [Mmlu-stem](#). 571

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits its reasoning in large language models. *Advances* 572
573
574
575

576 *in neural information processing systems*, 35:24824–
577 24837.

578 Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An,
579 Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xi-
580 aowei Lv, and 1 others. 2025. Light-r1: Curriculum
581 sft, dpo and rl for long cot from scratch and beyond.
582 *arXiv preprint arXiv:2503.10460*.

583 Yuyang Wu, Yifei Wang, Tianqi Du, Stefanie Jegelka,
584 and Yisen Wang. 2025. When more is less: Un-
585 derstanding chain-of-thought length in llms. *arXiv*
586 *preprint arXiv:2502.07266*.

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

590 Ling Yang, Zhaochen Yu, Bin Cui, and Mengdi
591 Wang. 2025. Reasonflux: Hierarchical llm reason-
592 ing via scaling thought templates. *arXiv preprint*
593 *arXiv:2502.06772*.

594 Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie
595 Xia, and Pengfei Liu. 2025. Limo: Less is more for
596 reasoning. *arXiv preprint arXiv:2502.03387*.

597 Huifeng Yin, Yu Zhao, Minghao Wu, Xuanfan Ni,
598 Bo Zeng, Hao Wang, Tianqi Shi, Liangying Shao,
599 Chenyang Lyu, Longyue Wang, and 1 others. 2025.
600 Towards widening the distillation bottleneck for rea-
601 soning models. *arXiv e-prints*, pages arXiv–2503.

602 Bin Yu, Hang Yuan, Haotian Li, Xueyin Xu, Yuliang
603 Wei, Bailing Wang, Weizhen Qi, and Kai Chen.
604 2025a. Long-short chain-of-thought mixture super-
605 vised fine-tuning eliciting efficient reasoning in large
606 language models. *arXiv preprint arXiv:2505.03469*.

607 Bin Yu, Hang Yuan, Yuliang Wei, Bailing Wang,
608 Weizhen Qi, and Kai Chen. 2025b. Long-short chain-
609 of-thought mixture supervised fine-tuning eliciting
610 efficient reasoning in large language models. *arXiv*
611 *preprint arXiv:2505.03469*.

612 Qianjin Yu, Keyu Wu, Zihan Chen, Chushu Zhang, Man-
613 lin Mei, Lingjun Huang, Fang Tan, Yongsheng Du,
614 Kunlin Liu, and Yurui Zhu. 2025c. Rethinking the
615 generation of high-quality cot data from the perspec-
616 tive of llm-adaptive question difficulty grading. *arXiv*
617 *preprint arXiv:2504.11919*.

618 Jintian Zhang, Yuqi Zhu, Mengshu Sun, Yujie Luo,
619 Shuofei Qiao, Lun Du, Da Zheng, Huajun Chen, and
620 Ningyu Zhang. 2025. [Lightthinker: Thinking step-
621 by-step compression](#). *CoRR*, abs/2502.15589.

622 Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying,
623 Liang He, and Xipeng Qiu. 2023. Evaluating the
624 performance of large language models on gaokao
625 benchmark. *arXiv preprint arXiv:2305.12474*.

626 Han Zhao, Haotian Wang, Yiping Peng, Sitong Zhao,
627 Xiaoyu Tian, Shuaiting Chen, Yunjie Ji, and Xian-
628 gang Li. 2025a. 1.4 million open-source distilled
629 reasoning dataset to empower large language model
630 training. *arXiv preprint arXiv:2503.19633*.

Shangziqu Zhao, Jiahao Yuan, Guisong Yang, and Us-
man Naseem. 2025b. Can pruning improve reason-
ing? revisiting long-cot compression with capabil-
ity in mind for better reasoning. *arXiv preprint*
arXiv:2505.14582.

A Complete Experiments

A.1 Exp-4: Generalization Analysis across Teacher Models

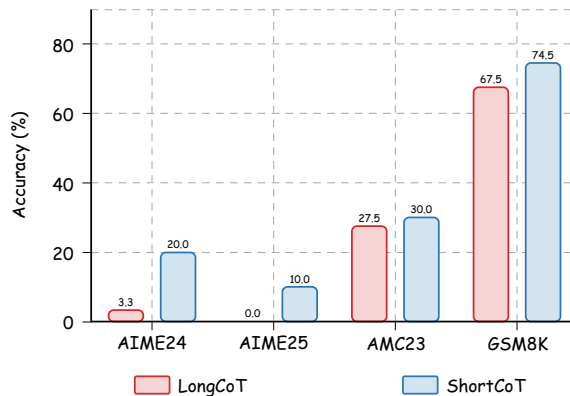


Figure 7: Performance comparison of LongCoT and ShortCoT distilled from the Qwen3-235B teacher model.

To verify that the effectiveness of our DAD method is not limited to a specific teacher model, we conducted a generalization experiment. In this experiment, we selected the most recent powerful Qwen3-235B-A22B-Thinking-2507 as the new teacher model. Using the same data sources as in **Exp-1**, we distilled ShortCoT by DAD to fine-tune the Qwen2.5-3B-Instruct base model. As illustrated in the figure 7, the results clearly indicate that the superiority of ShortCoT persists and remains significant even after changing the teacher model. Across all four mathematical benchmarks, the model trained on ShortCoT data achieved substantially higher accuracy than the one trained on LongCoT. This is especially pronounced on the challenging AIME24 task, where ShortCoT reached 20.0% accuracy compared to just 3.3% for LongCoT. Similarly, on the widely-used GSM8K dataset, ShortCoT demonstrated a commanding lead with 74% accuracy over LongCoT’s 67.5%. This outcome provides strong evidence for two conclusions:

- (1) Our DAD method possesses excellent generalization capabilities, as its effectiveness is not dependent on a specific teacher model.
- (2) Regardless of the teacher model’s strength, a verbose reasoning process itself is a suboptimal distillation strategy. Our proposed method of generating concise, key-step-focused reasoning chains

more efficiently transfers reasoning abilities to the student model.

A.2 Exp-5: Inference Efficiency Analysis: Token Savings Evaluation

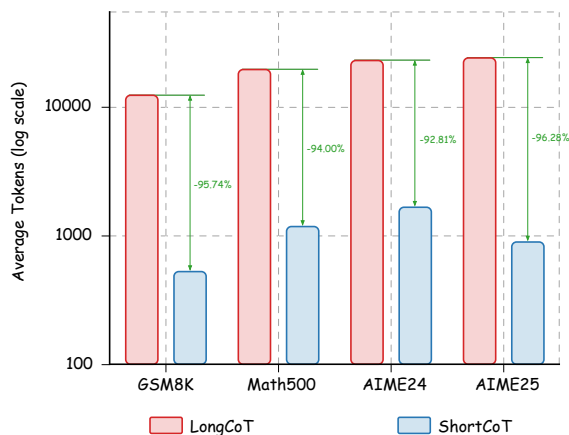


Figure 8: Comparison of Average Inference Tokens for LongCoT vs. ShortCoT across different benchmarks.

To quantify the improvements in inference efficiency offered by our DAP method, we analyzed the output token counts of the Qwen2.5-1.5B-Instruct model trained in **Exp-1**. This experiment was designed to measure the computational cost of the model during problem-solving. As illustrated in the figure above, the model trained on ShortCoT data, distilled via the DAD method, dramatically compresses inference costs.

Specifically, across all tested datasets, the average token output of the ShortCoT model was reduced by over 92% compared to the LongCoT model. For instance, on the GSM8K dataset, the token count plummeted from 12373 to 527, a saving of 95.74%. On AIME25, the token reduction was even more astounding, reaching 96.28%. This remarkable efficiency gain, combined with the findings from previous experiments where the ShortCoT model maintained or even surpassed the accuracy of the LongCoT model, proves the dual advantages of our approach. It not only guides the model to learn more precise and essential reasoning pathways but also drastically reduces the costs associated with model deployment and operation.

A.3 Stability of Difficulty Classification

To prove the stability of the difficulty classification of our pipeline, we ran an additional consistency test. We randomly sampled 50 questions, refined each long CoT independently 5 times using the DAD prompt, and set DeepSeek-R1 temperature to 1. Each refined CoT contains a fixed-format difficulty tag, so we obtained five tags per question

and checked full agreement. We find **88% full consistency** across the five refinements. The overall score is computed by first measuring agreement over the five runs per question and then aggregating across questions with weighting. This indicates that the teacher’s difficulty self-assessment under our prompt is reasonably stable, so the induced noise is limited. Even when a small number of labels are imperfect, they mainly affect the degree of shortening rather than the correctness of the distilled reasoning, because the hard-level rule explicitly requires keeping all necessary steps.

B Motivation Clarification

To address concerns regarding the perceived simplicity of our pipeline and the specific motivation for “human-like” reasoning, we clarify the design philosophy of the Difficulty-Aware Distillation (DAD) framework. We argue that DAD’s structured prompting is not merely a heuristic but a sophisticated routing paradigm designed to align LLM behavior with the efficient, pivotal-step-focused heuristics used by human experts.

- Clarification 1: Adaptive Prompting as a Reasoning Router.** We clarify that while DAD is implemented via prompting, it functions as a prompt-level router that dynamically allocates reasoning budgets based on task complexity. The paradigm of routing—directing inputs to specialized processing paths is a cornerstone of state-of-the-art models such as GPT-0SS. In our framework, we empower the LLM to act as its own autonomous router at the prompt level, ensuring that the student model learns the most efficient logical path instead of being forced into a rigid, one-size-fits-all verbosity.
- Clarification 2: Alignment with Human Reasoning Heuristics.** The motivation for “human-like” reasoning is rooted in the fact that true mathematical expertise is defined by the identification of pivotal logical nodes rather than the exhaustive listing of intermediate thoughts. Standard distillation methods often produce models that “overthink” simple problems, leading to high latency and error accumulation. By training on LiteCoT, we encourage the model to adopt a human-expert mindset—dedicating intense cognitive effort only to the “pivotal steps” of hard problems—thereby achieving a superior balance

754 between logical rigor and computational effi-
755 ciency.

756 C Difficulty Aware Prompt

757 This section provides the complete “Difficulty
758 Aware Prompt” used in the study. These prompts
759 are designed to guide the model in assessing the
760 difficulty of a problem and generating reasoning
761 procedures of appropriate length accordingly.

Difficulty Aware Prompt

As an assistant, you need to first assess the
↔ complexity of the problem
and adopt an appropriate thinking framework before
↔ providing the final
solution. Structure your response into two main
↔ sections: Thought and
Solution.

First evaluate the complexity of the problem, then
↔ choose a suitable
thinking framework, and describe the thought process
↔ as detailed as
possible:

1. For simple problems:

****Analysis:****
[Understand the core elements and goals of the
↔ problem]

****Approach:****
[Propose direct solution methods]

****Summary:****
[Concisely summarize the solution approach and key
↔ points]

2. For moderately complex problems:

****Analysis:****
[Understand the problem and identify key points and
↔ challenges]

****Initial Approach:****
[Propose preliminary solutions]

****Reflection:****
[Evaluate the pros and cons of the initial approach]

****Improvement:****
[Refine the solution based on reflection]

****Summary:****
[Summarize the key points of the final solution]

3. For highly complex problems:

****Analysis:****
[Analyze various aspects of the problem and break
↔ down its structure]

****Problem Decomposition:****
[Break complex problem into manageable sub-problems]

****Sub-problem Processing:**** (Repeat the following
↔ steps for each
sub-problem)

- Sub-problem 1:
 - * Initial approach
 - * Reflection
 - * Improved solution
- Sub-problem 2:
 - * Initial approach
 - * Reflection
 - * Improved solution

- ... (adjust according to the actual number of
↔ sub-problems)

****Integration:****
[Integrate sub-problem solutions into a complete
↔ solution]

****Overall Reflection:****
[Evaluate the integrated complete solution]

****Final Optimization:****
[Make final optimizations based on overall reflection]

****Summary:****
[Summarize key points of the final comprehensive
↔ solution]

The solution section should maintain logical,
↔ accurate, and concise
expression, detailing the steps needed to reach the
↔ conclusion,
formatted as:
****Solution:****
[Provide the final solution here]

763 D Extra Experiment Details 764

765 D.1 Training Details 766

767 Our training environment is equipped with 8 A800
768 GPUs with 80GB memory, and uses the Deep-
769 Speed ZeRO-3 framework for distributed training.
770 To further optimize GPU memory, we also intro-
771 duced technologies such as Tensor Parallelism and
772 liger kernel. Here, we share the YAML and Deep-
773 Speed configuration files used for qwen2.5-Math-
7B model training as an example.

Yaml config for running Qwen2.5-Math-7B on LiteCoT

```
# Logging and Output arguments
log_level: info
logging_strategy: steps
logging_steps: 10
save_total_limit: 1
report_to:
# - tensorboard
- wandb
save_strategy: steps
save_steps: 100
output_dir: /output_path
overwrite_output_dir: true

# Model arguments
model_name_or_path: /model_path
model_revision: main
trust_remote_code: True
torch_dtype: bfloat16
attn_implementation: sdpa

# Data training arguments
dataset_name: /dataset_path
dataset_format: parquet
# dataset_configs:
# - all
max_seq_length: 8192
dataset_num_proc: 8
packing: true

# SFT trainer config
preprocessing_num_workers: 8
seed: 233
```

762

774

```

do_train: true
num_train_epochs: 3
per_device_train_batch_size: 4
do_eval: true
eval_strategy: steps
eval_steps: 10
per_device_eval_batch_size: 1
optim: adamw_torch
learning_rate: 5.0e-5
lr_scheduler_type: cosine_with_min_lr
lr_scheduler_kwargs:
  min_lr_rate: 0.1
warmup_ratio: 0.1
weight_decay: 0.0001
gradient_accumulation_steps: 4
gradient_checkpointing: true
gradient_checkpointing_kwargs:
  use_reentrant: false
max_grad_norm: 1.0
bf16: true
use_liger: true

```

DeepSpeed config for running Qwen2.5-Math-7B on LiteCoT

```

compute_environment: LOCAL_MACHINE
debug: false
deepspeed_config:
  deepspeed_multinode_launcher: standard
  offload_optimizer_device: none
  offload_param_device: none
  zero3_init_flag: true
  zero3_save_16bit_model: true
  zero_stage: 3
  tensor_parallel:
    tp_size: 8
  offload_optimizer:
    device: none
    pin_memory: true
  offload_param:
    device: none
  train_batch_size: 16
  train_micro_batch_size_per_gpu: 4
  gradient_accumulation_steps: 4
  bf16:
    enabled: true
  optimizer:
    type: AdamW
    params:
      lr: 5.0e-5
      betas: [0.9, 0.999]
      eps: 1e-8
      weight_decay: 0.0001
      bias_correction: true
  main_process_port: 29500
  distributed_type: DEEPSPEED
  downcast_bf16: 'no'
  machine_rank: 0
  main_training_function: main
  mixed_precision: bf16
  num_machines: 1
  num_processes: 8
  rdzv_backend: static
  same_network: true
  tpu_env: []
  tpu_use_cluster: false
  tpu_use_sudo: false
  use_cpu: false

```

D.2 Evaluation Details

In our evaluation process, the evaluation scripts mainly come from two sources. For the four benchmarks AIME24, AIME25, GPQA, and MATH500, we use Hugging Face’s LightEval library <https://github.com/huggingface/>

[lighteval/tree/main](#) for model evaluation. For the remaining benchmarks focusing on mathematical reasoning ability, including GSM8K, Minerva Math, GaoKao2023en, OlympiadBench, College Math, Sat Math, MMLU STEM, and ACM23, we use the evaluation code provided in the Qwen2.5 code repository <https://github.com/QwenLM/Qwen2.5-Math/tree/main/evaluation/data>.

All evaluation experiments are performed uniformly under our specific GPU environment, NVIDIA H20 with 96GB memories.

All evaluation perform 5 times and use the highest score.

Next, we will share the shell (.sh) script files used to evaluate the AIME24, GPQA, MATH500, and AIME25 benchmarks based on Lighteval for reference.

Shell script for evaluating AIME24, AIME25, GPQA and MATH500

```

MODEL=/model_path
NUM_GPUS=n
export CUDA_VISIBLE_DEVICES="0,...,n-1"
MODEL_ARGS="pretrained=$MODEL,dtype=bfloat16,tensor]
↪ _parallel_size=$NUM_GPUS,\
max_model_length=32768,gpu_memory_utilization=0.8,\
generation_parameters={max_new_tokens:32768,tempera]
↪ ture:0.6,top_p:0.95}"
OUTPUT_DIR=data/evals/$MODEL

# AIME 2024
TASK=aime24
lighteval vllm $MODEL_ARGS "lighteval|$TASK|0|0" \
  --use-chat-template \
  --output-dir $OUTPUT_DIR

# AIME 2025
TASK=aime25
lighteval vllm $MODEL_ARGS "lighteval|$TASK|0|0" \
  --use-chat-template \
  --output-dir $OUTPUT_DIR

# MATH-500
TASK=math_500
lighteval vllm $MODEL_ARGS "lighteval|$TASK|0|0" \
  --use-chat-template \
  --output-dir $OUTPUT_DIR

# GPQA Diamond
TASK=gpqa:diamond
lighteval vllm $MODEL_ARGS "lighteval|$TASK|0|0" \
  --use-chat-template \
  --output-dir $OUTPUT_DIR

```

D.3 Benchmark Details

In our evaluation part, we totally evaluate our models on 11 benchmarks. They are:

- MATH500: Tests secondary school mathematics with open-ended problem-solving questions requiring step-by-step solutions, typically comprising 500 problems.
- AIME24: Assesses advanced high school mathematics (American Invitational Mathematics Examination level) through integer-

811	answer competition-style problems, with a	across various specific STEM subjects (e.g.,	858
812	specific recent contest (e.g., AIME 2024) hav-	the "Mathematics" part alone has 250 dev and	859
813	ing 30 questions.	1175 test questions).	860
814	• GPQA: Evaluates graduate-level physics	• ACM23: Assesses competitive programming	861
815	quantitative reasoning with challenging, multi-	skills (similar to ACM International Colle-	862
816	step, free-response questions written by	giate Programming Contest problems) with al-	863
817	physics experts, containing 448 questions.	gorithmic reasoning challenges requiring code	864
818	• GSM8K: Focuses on grade school mathemat-	solutions or detailed algorithmic steps, with	865
819	ics (grades 3-8) with multi-step arithmetic	specific contest sets varying but problem col-	866
820	word problems requiring basic operations, fea-	lections often containing hundreds of prob-	867
821	tureing approximately 1,319 test questions.	lems (the user might be referring to a specific	868
822	• Minerva Math: Covers a broad range of quan-	year's problem set or a curated collection).	869
823	titative reasoning from elementary mathemat-	E Experiment Setup	870
824	ics to advanced levels using free-response	We conduct extensive experiments to evaluate the	871
825	math problems, often drawing from existing	effectiveness of LiteCoT data. Our evaluations ad-	872
826	datasets and typically used with around 2,000	dress three key questions: (Exp-1) Does training on	873
827	questions from its test set (MATH subset).	short difficulty-aware CoTs, LiteCoT, actually im-	874
828	• GaoKao2023: Tests Chinese National College	prove model performance compared to long CoTs?	875
829	Entrance Examination (Gaokao) mathematics	(Exp-2) Can our DAP method rival or surpass mod-	876
830	questions from 2023, featuring complex prob-	els distilled with an order of magnitude more data	877
831	lems covering a wide high-school curriculum,	from a powerful teacher? (Exp-3) How does our	878
832	with the math section typically having around	DAP approach compare to other recent CoT distil-	879
833	20-22 questions per exam paper version.	lation and compression techniques?	880
834	• OlympiadBench: Assesses mathematical	E.1 Experiment Setup	881
835	Olympiad problem-solving abilities with	Training Data. Our experimental evaluation con-	882
836	highly challenging, proof-oriented or complex	sists of three stages. All data comes from the	883
837	computational questions from various interna-	OpenThoughts (Team, 2025b) dataset and is pro-	884
838	tional and national Olympiads, containing 185	cessed using our DAP distillation process. For	885
839	problems.	Exp-1: Use 25K OpenThoughts samples. Their	886
840	• College Math: Covers US college-level math-	original Long CoT form is used as the baseline	887
841	ematics topics with multiple-choice and free-	training data; at the same time, these samples are	888
842	response questions spanning calculus, algebra,	processed by the DAP process to generate 25K cor-	889
843	and statistics, with one common dataset hav-	responding Short CoT data for comparison training.	890
844	ing 1,760 questions.	For Exp-2: Expand the amount of OpenThoughts	891
845	• Sat Math: Evaluates high school mathemat-	data to 100K. These samples are processed by the	892
846	ics relevant to college readiness (SAT exam)	DAP process to generate 100K short CoT data(<i>i.e.</i> ,	893
847	through multiple-choice and grid-in questions	LiteCoT), which constitute the training set of this	894
848	on algebra, problem-solving, and data analy-	experiment. For Exp-3: Directly use the 25K short	895
849	sis, with the math sections of an official SAT	CoT data generated in Exp-1 as the training data	896
850	practice test typically containing 58 questions.	for this stage.	897
851	• MMLU STEM: A subset of the Massive	Base Models. In Exp-1 and Exp-3 , due to the	898
852	Multitask Language Understanding (MMLU)	small amount of training data, the models we se-	899
853	benchmark, it tests expertise in STEM fields	lected are all instruction versions in order to ob-	900
854	(Science, Technology, Engineering, and Math-	tain better instruction following capabilities on	901
855	ematics) at high school and college levels via	small datasets. For Exp-1: Qwen2.5-0.5B-Instruct,	902
856	multiple-choice questions, with the STEM	Qwen2.5-Math-1.5B-Instruct, Qwen2.5-Math-7B-	903
857	subset comprising several thousand questions	Instruct, Qwen2.5-14B-Instruct of the Qwen2.5	904
		series, and Llama3.1-8B-instruct of the Llama3.1	905

series are included as base models. For **Exp-2**: Aiming to make a fair comparison with the results in the DeepSeek R1 report, we selected the same base version models used in the report: Qwen2.5-math-1.5B, Qwen2.5-math-7B, Qwen2.5-14B, and Qwen2.5-32B. For **Exp-3**: Similar to Exp-1, the model Qwen2.5-math-7B-Instruct is selected as the baseline model.

Benchmark Datasets. We evaluate model accuracy on a diverse suite of 11 benchmarks, covering math word problems (MATH500 (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021)), competition-level math (AIME24 (AI-MO, 2024)/AIME25 (AI-MO, 2025), OlympiadBench (Choquette-Choo et al., 2023), MinervaMath (math ai, 2025)), general QA (GPQA) (Rein et al.), and academic exams (GaoKao2023 (Zhang et al., 2023), MMLU-STEM (TIGER-Lab, 2024), etc.).

Baselines. For **Exp-1**, we mainly use the model directly distilled from the original Long CoT as the baseline to evaluate the relative improvement of our method. For **Exp-2**, we include a series of current mainstream reasoning models in the benchmark comparison range, including representative models such as S1 (Muennighoff et al., 2025) and OpenThoughts (Team, 2025b). For **Exp-3**, our benchmark comparison focuses on different Long CoT optimization or “decompose” methods. These methods include strategies that optimize at the prompt engineering level, such as Chain-of-Draft (Xu et al., 2025) and Budget Aware Reasoning (Han et al., 2024); in addition, we also include methods that require training additional models to simplify the output content, such as LLMLingua2 (Pan et al., 2024).

F Dataset Comparison

F.1 Cases of Samples

In this section, we will show the model distillation results for the same problem in different datasets. We will show the comparison of three different difficulty levels: easy, medium, and hard. Because the original answer for hard is too long, we only show the Litecot data and OpenThoughts data for easy and medium difficulty levels here.

Easy Question:

Return your final response within `\boxed{}`.
Find all natural numbers that,
when divided by 7, give the same quotient and remainder.

Solutions of easy problems in LiteCoT

```
<|begin_of_thought|>
**Analysis:**
The problem requires finding all natural numbers  $n$ 
 $\rightarrow$   $n$  such that
when divided by 7, the quotient  $q$  equals the
 $\rightarrow$  remainder
 $\rightarrow$   $r$ . Using the division algorithm,  $n = 7q + r$ 
 $\rightarrow$   $n$ ,
with  $0 \leq r < 7$ . The condition  $q = r$ 
 $\rightarrow$  must hold.
```

```
**Approach:**
Substitute  $q = r$  into the division equation to
 $\rightarrow$  get  $n = 8r$ .
Since  $r$  must be a valid remainder ( $1 \leq r$ 
 $\rightarrow$   $r \leq 6$ ,
as  $r = 0$  would make  $n = 0$ , not a natural
 $\rightarrow$  number), compute
 $\rightarrow$   $n$  for  $r = 1, 2, \dots, 6$ .
```

```
**Summary:**
The natural numbers are  $8r$  for  $r = 1$  to  $6$ ,
 $\rightarrow$  yielding
 $\rightarrow$   $\{8, 16, 24, 32, 40, 48\}$ .
<|end_of_thought|>
```

```
<|begin_of_solution|>
**Solution:**
To find all natural numbers  $n$  such that when
 $\rightarrow$  divided by 7,
the quotient equals the remainder:
1. Use the division formula:  $n = 7q + r$ , where
 $\rightarrow$   $0 \leq r < 7$ .
2. Set  $q = r$ , leading to  $n = 7r + r = 8r$ .
3. Since  $r$  must be a natural number (remainder
 $\rightarrow$   $\geq 1$ ) and
 $\rightarrow$   $r < 7$ , valid values for  $r$  are  $\{1, 2, 3,$ 
 $\rightarrow$   $4, 5, 6\}$ .
4. Calculate  $n$  for each  $r$ :
-  $r = 1 \rightarrow n = 8$ 
-  $r = 2 \rightarrow n = 16$ 
-  $r = 3 \rightarrow n = 24$ 
-  $r = 4 \rightarrow n = 32$ 
-  $r = 5 \rightarrow n = 40$ 
-  $r = 6 \rightarrow n = 48$ 
```

```
Thus, the natural numbers are:
 $\rightarrow$   $\boxed{\{8, 16, 24, 32, 40, 48\}}$ 
<|end_of_solution|>
```

951

Solutions of easy problems in OpenThoughts

```
<|begin_of_thought|>
Okay, so I need to find all natural numbers that,
 $\rightarrow$  when divided by 7,
give the same quotient and remainder. Let me start by
 $\rightarrow$  understanding
the problem properly.
```

```
First, let's recall that when we divide a natural
 $\rightarrow$  number by 7,
we can express it in the form of the division
 $\rightarrow$  algorithm:  $n = 7q + r$ ,
where  $q$  is the quotient and  $r$  is the remainder. The
 $\rightarrow$  remainder  $r$  must
satisfy  $0 \leq r < 7$  because we're dividing by 7. So,
 $\rightarrow$  the possible
values for  $r$  are  $0, 1, 2, 3, 4, 5, 6$ .
```

```
Now, the problem states that the quotient and the
 $\rightarrow$  remainder are the same.
That means  $q = r$ . So substituting that into the
 $\rightarrow$  equation,
we get  $n = 7q + q = 8q$ . Therefore,  $n$  must be a
 $\rightarrow$  multiple of 8. But wait,
```

952

we need to ensure that the remainder r is indeed
↪ equal to q and that it
satisfies the condition $0 \leq r < 7$.

Since $r = q$, and r must be less than 7,
 q must also be less than 7. Because if q were 7 or
↪ more,
then the remainder r would be 7 or more, which isn't
↪ allowed in division
by 7. So q can be 0, 1, 2, 3, 4, 5, or 6.

However, n is a natural number. Depending on the
↪ definition,
natural numbers usually start from 1, so n must be at
↪ least 1. If $q = 0$,
then $n = 8 \times 0 = 0$, which might not be considered a
↪ natural number.
Let me check if 0 is included here. If the problem
↪ defines natural
numbers as starting from 1, then q can only be from 1
↪ to 6. But if 0 is
included, then 0 would also be a solution.

But let me confirm the standard definition.
In mathematics, natural numbers typically start from
↪ 1, so 0 is excluded.
Therefore, q must be at least 1. So q can be 1, 2, 3,
↪ 4, 5, 6. Therefore,
the corresponding n values would be $8 \times 1 = 8$, $8 \times 2 = 16$,
↪ $8 \times 3 = 24$, $8 \times 4 = 32$,
 $8 \times 5 = 40$, $8 \times 6 = 48$.

But wait, I need to verify if each of these numbers
↪ actually satisfies
the original condition. Let's take each n and divide
↪ by 7 to see if the
quotient and remainder are equal.

Starting with $n=8$: 8 divided by 7 is 1 with a
↪ remainder of 1.
Here, quotient $q=1$ and remainder $r=1$, so they are
↪ equal. That works.

Next, $n=16$: 16 divided by 7 is 2 with a remainder of
↪ 2. So $q=2$ and $r=2$.
That's good.

$n=24$: 24 divided by 7 is 3 with a remainder of 3.
↪ Because $7 \times 3 = 21$,
 $24 - 21 = 3$. So $q=3$, $r=3$. Correct.

$n=32$: 32 divided by 7 is 4 with remainder 4 ($7 \times 4 = 28$,
↪ $32 - 28 = 4$).
So $q=4$, $r=4$.

$n=40$: 40 divided by 7 is 5 with remainder 5. $7 \times 5 = 35$,
↪ $40 - 35 = 5$.
So $q=5$, $r=5$.

$n=48$: 48 divided by 7 is 6 with remainder 6. $7 \times 6 = 42$,
↪ $48 - 42 = 6$.
So $q=6$, $r=6$.

Okay, all of these check out. Now, if we consider $q=7$,
↪ then $n=8 \times 7 = 56$.
Let's divide 56 by 7: quotient is 8, remainder is 0.
↪ Here, q would have
been 8 (but according to our original substitution,
↪ $q=r=7$). Wait,
this is where the problem arises. If $q=7$, then $r=7$,
↪ but 7 is not a valid
remainder when dividing by 7. The remainder must be
↪ less than 7.
Therefore, $q=7$ is invalid because it would lead to a
↪ remainder of 7,
which isn't allowed. Hence, the maximum q can be is 6.

Therefore, the natural numbers are 8, 16, 24, 32, 40,
↪ 48.

Wait a second, let me make sure there are no other
↪ numbers beyond these
that could satisfy the condition. Suppose someone
↪ claims that there is a

larger number where $q=r$. Let's say $q=7$, but as I saw,
↪ r would have to be
7, which is not allowed. Similarly, for $q=8$, $r=8$
↪ which is even more
invalid. So the only possible q values are 1 through
↪ 6,
leading to $n=8q$ where $q=1$ to 6.

But just to be thorough, let's check if there's a
↪ different approach.
Suppose we set up the equation:

When n is divided by 7, quotient q equals remainder r .
So, by division algorithm:

$$n = 7q + r$$

But since $q = r$, substitute:

$$n = 7q + q = 8q$$

Additionally, the remainder r must satisfy $0 \leq r <$
↪ 7 . Since $r = q$,
this translates to $0 \leq q < 7$. However, since n is a
↪ natural number
(assuming starting at 1), q must be at least 1.
↪ Therefore, q
{1,2,3,4,5,6}, leading to n {8,16,24,32,40,48}.

Therefore, these are all the natural numbers that
↪ satisfy the condition.
Let me check if $n=0$ is allowed. If the problem
↪ includes 0 as a natural
number, which is sometimes the case in certain
↪ contexts, then 0 would be
a solution as well (0 divided by 7 is 0 with remainder
↪ 0). However,
since the problem specifies "natural numbers," and
↪ given that in most
mathematical contexts natural numbers start from 1,
↪ we can exclude 0.

Thus, the final answer should be the numbers 8, 16,
↪ 24, 32, 40, and 48.

<|end_of_thought|>

<|begin_of_solution|>

To find all natural numbers (n) that, when
↪ divided by 7, yield the
same quotient (q) and remainder (r) , we start
↪ with the division
algorithm:
 $[n = 7q + r]$
where $(0 \leq r < 7)$. The problem states $(q = r)$,
↪ so substituting $(r = q)$ gives:
 $[n = 7q + q = 8q.]$

Since $(r = q)$ must satisfy $(0 \leq q < 7)$,
and (n) is a natural number (starting from 1),
 (q) must be in $(\{1, 2, 3, 4, 5, 6\})$.
Substituting these values into $(n = 8q)$:

- For $(q = 1)$: $(n = 8 \times 1 = 8)$.
- For $(q = 2)$: $(n = 8 \times 2 = 16)$.
- For $(q = 3)$: $(n = 8 \times 3 = 24)$.
- For $(q = 4)$: $(n = 8 \times 4 = 32)$.
- For $(q = 5)$: $(n = 8 \times 5 = 40)$.
- For $(q = 6)$: $(n = 8 \times 6 = 48)$.

Verification confirms each (n) satisfies $($
↪ $\text{quotient} =$
 $\text{remainder})$ when divided by 7. For example:
- $(8 \div 7 = 1)$ with remainder (1) ,
- $(16 \div 7 = 2)$ with remainder (2) , etc.

****Final Answer:****

The natural numbers are $(\boxed{8, 16, 24, 32, 40,$
↪ $48})$.

<|end_of_solution|>

Medium Question:

Return your final response within $\boxed{}$.
 Let (h_1, h_2, h_3) be the heights of a triangle,
 and (r) be the radius of the inscribed circle.
 Prove that $(h_1 + h_2 + h_3) \geq 9r$.

Solutions of medium problems in LiteCoT

\langle begin_of_thought \rangle

Analysis:

The problem requires proving an inequality involving
 \rightarrow triangle heights
 and inradius. Key elements are the relationships
 \rightarrow between heights $(h_1,$
 $h_2, h_3)$, inradius (r) , area (S) , and semiperimeter
 \rightarrow (p) . The challenge
 lies in connecting these quantities through known
 \rightarrow formulas and applying
 an inequality.

Initial Approach:

Express heights in terms of area and sides $(h_i =$
 $\rightarrow 2S/a_i)$. Use the
 identity $S = p \cdot r$ to substitute into the sum of
 \rightarrow heights. This leads to
 $h_1 + h_2 + h_3 = 2pr(1/a + 1/b + 1/c)$. To establish
 \rightarrow the inequality,
 relate $1/a + 1/b + 1/c$ to semiperimeter via
 \rightarrow Cauchy-Schwarz.

Reflection:

The Cauchy-Schwarz inequality (for vectors (a, b, c)
 \rightarrow and $(1/a, 1/b, 1/c)$)
 provides $(a + b + c)(1/a + 1/b + 1/c) \geq 9$. Since a
 \rightarrow $+ b + c = 2p$, this
 bounds $1/a + 1/b + 1/c \geq 9/(2p)$. Substituting back
 \rightarrow confirms
 the required inequality.

Improvement:

Verify the validity of Cauchy-Schwarz application and
 \rightarrow ensure all
 substitutions align with triangle properties. No
 \rightarrow logical gaps detected.

Summary:

The proof hinges on expressing heights via area,
 \rightarrow applying Cauchy-Schwarz
 to relate reciprocals of sides to semiperimeter, and
 \rightarrow substituting into
 the sum of heights to derive the inequality.
 \langle end_of_thought \rangle

\langle begin_of_solution \rangle

Solution:

1. ****Express heights via area**:** For a triangle with
 \rightarrow sides (a, b, c)
 and corresponding heights (h_1, h_2, h_3) , the area
 \rightarrow (S) is:

$$\begin{aligned} h_1 &= \frac{2S}{a}, \quad h_2 = \frac{2S}{b}, \quad h_3 = \frac{2S}{c}. \end{aligned}$$

2. ****Sum of heights**:**

$$\begin{aligned} h_1 + h_2 + h_3 &= 2S \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \\ &\geq 2S \cdot \frac{9}{2p} \end{aligned}$$

3. ****Relate (S) to inradius (r) **:**

Using $(S = p \cdot r)$ (where $(p = \frac{a+b+c}{2})$
 \rightarrow is semiperimeter):

$$\begin{aligned} h_1 + h_2 + h_3 &= 2pr \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \\ &\geq 2pr \cdot \frac{9}{2p} \\ &\geq 9r \end{aligned}$$

4. ****Apply Cauchy-Schwarz inequality**:**

$$\begin{aligned} (a + b + c) \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) &\geq (1 + 1 + 1)^2 = 9. \end{aligned}$$

Since $(a + b + c = 2p)$, this simplifies to:

$$\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \geq \frac{9}{2p}.$$

5. ****Substitute into height sum**:**

$$h_1 + h_2 + h_3 \geq 2pr \cdot \frac{9}{2p} = 9r.$$

Thus, the inequality $(h_1 + h_2 + h_3 \geq 9r)$ is
 \rightarrow proven.

$$\boxed{h_1 + h_2 + h_3 \geq 9r}$$

\langle end_of_solution \rangle

957

Solutions of medium problems in OpenThoughts

\langle begin_of_thought \rangle

Alright, so I need to prove that the sum of the
 \rightarrow heights of a triangle
 is at least nine times the inradius. Let me start by
 \rightarrow recalling some
 formulas related to triangles, heights, and the
 \rightarrow inradius.

First, the height (altitude) corresponding to a side
 \rightarrow of a triangle
 is given by the formula $(h_a = \frac{2\Delta}{a})$,
 \rightarrow where
 (Δ) is the area of the triangle and (a)
 \rightarrow is the length
 of
 the side. Similarly, $(h_b = \frac{2\Delta}{b})$ and
 $(h_c = \frac{2\Delta}{c})$. So, the sum of the
 \rightarrow heights
 $(h_1 + h_2 + h_3)$ would be $(\frac{2\Delta}{a} + \frac{2\Delta}{b} + \frac{2\Delta}{c})$.

On the other hand, the inradius (r) is related to
 \rightarrow the area and
 the
 semiperimeter (s) by the formula $(r =$
 $\rightarrow \frac{\Delta}{s})$,
 where
 $(s = \frac{a + b + c}{2})$. So, the inradius is the
 \rightarrow area divided
 by
 the semiperimeter.

Our goal is to show that $(h_1 + h_2 + h_3 \geq 9r)$
 \rightarrow . Let's
 substitute
 the expressions for the heights and the inradius in
 \rightarrow terms of the
 area
 and the sides.

First, express $(h_1 + h_2 + h_3)$ as $(2\Delta$
 $\rightarrow \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right))$. The inradius
 \rightarrow (r) is
 $(\frac{\Delta}{s})$, so $(9r = \frac{9\Delta}{s})$
 \rightarrow .
 Therefore, the inequality we need to prove becomes:

$$2\Delta \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq \frac{9\Delta}{s}$$

$\frac{9\Delta}{s}$ \)

Since Δ is positive (it's the area of a triangle), we can divide both sides by Δ without changing the inequality direction:

$$2 \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq \frac{9}{s}$$

Simplifying further, multiply both sides by s :

$$2s \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9$$

So, the inequality reduces to proving that $2s \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9$, where $s = \frac{a+b+c}{2}$.

Let me write this more explicitly. Substitute $s = \frac{a+b+c}{2}$, so:

$$2 \times \frac{a+b+c}{2} \times \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9$$

The 2 and the denominator 2 cancel out, so:

$$(a+b+c) \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9$$

Ah, now this looks familiar. This is a classic inequality that might be related to the AM-HM inequality or Cauchy-Schwarz. Let me recall the Cauchy-Schwarz inequality in the form:

$$(x_1^2 + x_2^2 + x_3^2)(y_1^2 + y_2^2 + y_3^2) \geq (x_1y_1 + x_2y_2 + x_3y_3)^2$$

But perhaps another form is more useful here. The Cauchy-Schwarz inequality for sequences (a, b, c) and $\left(\frac{1}{a}, \frac{1}{b}, \frac{1}{c} \right)$ gives:

$$\left(a + b + c \right) \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq \left(\frac{a^2}{a} + \frac{b^2}{b} + \frac{c^2}{c} \right)$$

But this simplifies to $(1+1+1)^2 \leq (a^2 + b^2 + c^2) \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right)$, which is $9 \leq$ something. Not sure if that's directly helpful here.

Alternatively, consider the AM-HM inequality for three positive real numbers:

$$\frac{a+b+c}{3} \geq \frac{3}{\frac{1}{a} + \frac{1}{b} + \frac{1}{c}}$$

Rearranging this gives:

$$(a+b+c) \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9$$

Which is exactly the inequality we need! So, according to the AM-HM inequality,

the product of the sum of the sides and the sum of their reciprocals is indeed greater than or equal to 9. Therefore, the original inequality $h_1 + h_2 + h_3 \geq 9r$ holds true.

But wait, let me double-check the steps to make sure I didn't skip anything crucial. Starting from the heights:

$h_1 = \frac{2\Delta}{a}$, similarly for h_2, h_3 . Then sum is $2\Delta \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right)$. Then, $r = \frac{\Delta}{s}$, so $9r = \frac{9\Delta}{s}$. Therefore, the inequality becomes:

$$2\Delta \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq \frac{9\Delta}{s}$$

Cancel Δ :

$$2 \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq \frac{9}{s}$$

Multiply both sides by s :

$$2s \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9$$

But since $s = \frac{a+b+c}{2}$, substituting:

$$2 \times \frac{a+b+c}{2} \times \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9$$

Which is exactly the AM-HM inequality. So, yes, the steps are correct.

But hold on, AM-HM gives $\frac{a+b+c}{3} \geq \frac{3}{1/a + 1/b + 1/c}$, which rearranges to $(a+b+c) \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9$. So that's exactly the inequality we have here. Therefore, equality holds when $a = b = c$, i.e., in an equilateral triangle. So, this inequality is tight for equilateral triangles.

Therefore, the original inequality $h_1 + h_2 + h_3 \geq 9r$ is proven by applying the AM-HM inequality to the sides of the triangle.

Alternatively, another approach might involve expressing the heights in terms of the inradius and other elements, but since this path worked out, maybe that's sufficient. However, to be thorough, let me consider if there are other ways or possible pitfalls.

Wait, another thought: The formula for the inradius is $r = \frac{\Delta}{s}$, so $s = \frac{\Delta}{r}$. Then, since $h_1 + h_2 + h_3 = 2\Delta \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right)$, we can write the inequality as:

$$2\Delta \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9 \frac{\Delta}{s}$$

Cancelling Δ , we get $2 \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq \frac{9}{s}$, which is equivalent to $2s \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9$. Then substituting $s = \frac{a+b+c}{2}$, we have:

$$2 \times \frac{a+b+c}{2} \times \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9$$

Which again is the same inequality. So, regardless of the path, we end up with the AM-HM inequality.

Alternatively, if I didn't recall AM-HM, maybe using
 \hookrightarrow Cauchy-Schwarz:

$$\sqrt{\left(\frac{1}{\sqrt{a}}\right)^2 + \left(\frac{1}{\sqrt{b}}\right)^2 + \left(\frac{1}{\sqrt{c}}\right)^2} \cdot \sqrt{a^2 + b^2 + c^2} \geq a + b + c$$

But that's the same as $(a + b + c)(1/a + 1/b + 1/c) \geq 9$. So, Cauchy-Schwarz also gives the same result. Therefore, the inequality holds by Cauchy-Schwarz as well.

Therefore, regardless of the method used, the key inequality needed is $(a + b + c)(1/a + 1/b + 1/c) \geq 9$, which is a standard result. Therefore, the original inequality is proven.

Another angle: Let's consider specific cases. For an equilateral triangle, let's check if equality holds. Suppose all sides are equal, say length a . Then, the semiperimeter $s = 3a/2$. The inradius $r = \Delta/s$. The area $\Delta = (\sqrt{3}/4)a^2$. So $r = (\sqrt{3}/4)a$. The heights h_1, h_2, h_3 are all equal to $(\sqrt{3}/2)a$. So sum of heights is $3 \times (\sqrt{3}/2)a = (3\sqrt{3}/2)a$. Then $9r = 9 \times (\sqrt{3}/4)a = (9\sqrt{3}/4)a$. Therefore, sum of heights equals $9r$ in this case, so equality holds.

For a non-equilateral triangle, let's take a simple case. Suppose $a = 3, b = 4, c = 5$ (right-angled triangle). Then, semiperimeter $s = (3 + 4 + 5)/2 = 6$. Area $\Delta = (3 \times 4)/2 = 6$. Inradius $r = \Delta/s = 6/6 = 1$. Heights: $h_1 = 2\Delta/a = 12/3 = 4, h_2 = 12/4 = 3, h_3 = 12/5 = 2.4$. Sum of heights: $4 + 3 + 2.4 = 9.4$. So $9.4 \geq 9$.

Another example: $a = 2, b = 2, c = 3$. Then $s = (2 + 2 + 3)/2 = 3.5$. Area using Heron's formula: $\sqrt{3.5(3.5 - 2)(3.5 - 2)(3.5 - 3)} = \sqrt{3.5 \times 1.5 \times 1.5 \times 0.5} = \sqrt{3.5 \times 1.5 \times 0.75} = \sqrt{3.9375} \approx 1.984$. So $r = 1.984 / 3.5 \approx 0.567$. Heights: $h_1 \approx 2 \times 1.984 / 2 \approx 1.984, h_2 \approx 1.984, h_3 \approx 2 \times 1.984 / 3 \approx 1.323$. Sum $\approx 1.984 + 1.984 + 1.323 \approx 5.291$. So $5.291 \geq 5.103$.

Wait, but according to the calculation, the sum of heights is approximately 5.291 , which is greater than $9r \approx 5.103$. So, even in this case, the inequality holds.

But let me check a more extreme triangle. Suppose a triangle with sides approaching degeneracy. Let's take $a = 1, b = 1, c$ approaching 2 (but still a valid triangle). Let's compute when c approaches 2 , say $c = 2 - \epsilon$, where ϵ is very small. Then semiperimeter $s = (1 + 1 + 2 - \epsilon)/2 = (4 - \epsilon)/2 = 2 - \epsilon/2$. Area Δ using Heron's formula: $\sqrt{s(s-a)(s-b)(s-c)} = \sqrt{(2 - \epsilon/2)(1 - \epsilon/2)(1 - \epsilon/2)(\epsilon/2)}$. As $\epsilon \rightarrow 0$, this becomes $\sqrt{(2)(1)(1)(0)} = 0$. So $\Delta \rightarrow 0$, and $r = \Delta/s \rightarrow 0$. The heights: $h_1 = 2\Delta/a = 2\Delta, h_2 = 2\Delta, h_3 = 2\Delta/c \approx 2\Delta$.

So sum of heights $\approx 2\Delta + 2\Delta + \Delta = 5\Delta$. Since $5 > 4.5$, $5\Delta \geq (9/2)\Delta$ when $\Delta > 0$. As $\Delta \rightarrow 0$, both sides $\rightarrow 0$, inequality still holds.

Alternatively, consider a very skewed triangle where one side is much longer than the others. But perhaps in all cases, the sum of the heights is still greater than $9r$.

Therefore, it seems that through multiple examples and applying the AM-HM inequality, the original statement is valid.

Another approach could be using trigonometric identities. Let's denote the triangle with sides (a, b, c) opposite to angles (A, B, C) respectively. The heights can be expressed as $h_a = b \sin C = c \sin B$, similarly for others. But this might complicate things. Alternatively, since $r = (a + b - c)/2 \tan(C/2)$ etc., but perhaps this is not useful here.

Alternatively, express everything in terms of r and the semiperimeter. Since $r = \Delta/s$, and $\Delta = rs$. Then, the heights can be written as $h_a = 2\Delta/a = 2rs/a$. Similarly, $h_b = 2rs/b, h_c = 2rs/c$. Therefore, sum of heights is $2rs(1/a + 1/b + 1/c) \geq 9r$. We need to show that this is $\geq 9r$. Divide both sides by r (positive), so need to show $2s(1/a + 1/b + 1/c) \geq 9$, which is the same as before. So again, we come back to the same inequality.

Therefore, regardless of the approach, the crux is proving $(a + b + c)(1/a + 1/b + 1/c) \geq 9$, which is a direct consequence of the AM-HM inequality. Hence, the original inequality holds.

- In conclusion, the key steps are:
- Express the heights in terms of the area and sides.
 - Express the inradius in terms of the area and semiperimeter.
 - Formulate the inequality in terms of the sides and semiperimeter.
 - Apply the AM-HM inequality to establish the required result.
 - Verify with specific examples to ensure validity.

Therefore, the inequality $h_1 + h_2 + h_3 \geq 9r$ is proven.

</end_of_thought|>
 </begin_of_solution|>

To prove that the sum of the heights of a triangle $h_1 + h_2 + h_3$ is at least nine times the inradius r , we proceed as follows:

- Express Heights and Inradius:**
 - Heights: $h_a = \frac{2\Delta}{a}, h_b = \frac{2\Delta}{b}, h_c = \frac{2\Delta}{c}$, where Δ is the area and (a, b, c) are the sides.
 - Sum of heights: $h_1 + h_2 + h_3 = 2\Delta \left(\frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right)$.
 - Inradius: $r = \frac{\Delta}{s}$, where $s = \frac{a + b + c}{2}$ is the semiperimeter.

```

2. Formulate the Inequality:
   Substitute  $(h_1 + h_2 + h_3)$  and  $(9r)$  into
   the inequality:
   \[
   2\Delta \left( \frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq \frac{9\Delta}{s}.
   \]
   Cancel  $(\Delta)$  and multiply through by  $(s)$ :
   \[
   2s \left( \frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9.
   \]

3. Simplify Using Semiperimeter:
   Substitute  $(s = \frac{a+b+c}{2})$ :
   \[
   (a+b+c) \left( \frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9.
   \]
   which rearranges to:
   \[
   (a+b+c) \left( \frac{1}{a} + \frac{1}{b} + \frac{1}{c} \right) \geq 9.
   \]
   Equality holds if and only if  $(a = b = c)$ 
   (equilateral triangle).

5. Conclusion:
   The inequality  $(h_1 + h_2 + h_3 \geq 9r)$ 
   follows directly from the AM-HM inequality
   applied to the sides of the triangle. Examples
   and edge cases confirm the validity of the
   result.

Final Answer:

$$\boxed{h_1 + h_2 + h_3 \geq 9r}$$

</end_of_solution>

```

G Example Output

In this chapter, we demonstrate the differences in output between the model distilled using our dataset and the model distilled using LongCoT when applied to the same problem. To be specific, the models used in this section are Qwen2.5-7B-Instruct(short) and Qwen2.5-7B-Instruct(long) from Exp-1.

Example Question:

Let $\triangle ABC$ be a triangle inscribed in circle ω . Let the tangents to ω at B and C intersect at point D , and let \overline{AD} intersect ω at P . If $AB=5$, $BC=9$, and $AC=10$, AP can be written as the form $\frac{m}{n}$, where m and n are relatively prime integers. Find $m+n$.

Output from Qwen2.5-7B-Instruct(short)

```

<|begin_of_thought|>
Analysis:
The problem involves triangle  $(ABC)$  inscribed in a circle  $(\omega)$ , with tangents at  $(B)$  and  $(C)$  intersecting at  $(D)$ . Line  $(AD)$  intersects  $(\omega)$  again at  $(P)$ . Given side lengths  $(AB = 5)$ ,  $(BC = 9)$ , and  $(AC = 10)$ , we need to find  $(AP = \frac{m}{n})$  and compute  $(m+n)$ .

Problem Decomposition:
1. Power of a Point at  $(D)$ : Use  $(DB^2 = DP \cdot DA)$  to relate  $(AP)$  to other segments.
2. Coordinate Geometry: Compute coordinates of  $(B)$ ,  $(C)$ , and  $(D)$  to find the equation of  $(AD)$  and its intersection  $(P)$  with  $(\omega)$ .
3. Law of Cosines and Sines: Calculate  $(\cos A)$ ,  $(\cos B)$ , and  $(\cos C)$  to derive  $(AP)$  via trigonometric identities and circle properties.

Sub-problem Processing:
- Sub-problem 1 (Power of a Point):
  - Initial Approach: Use  $(DB^2 = DP \cdot DA)$ . Requires  $(DB)$  and  $(AD)$ .
  - Reflection:  $(DB)$  depends on coordinates of  $(D)$ , which is complex without coordinates.
  - Improved Solution: Compute coordinates of  $(D)$  via tangents and solve for  $(AD)$  and  $(DP)$ .

- Sub-problem 2 (Coordinate Geometry):
  - Initial Approach: Place  $(A)$  at  $((0,0))$ ,  $(B)$  at  $((5,0))$ , and compute  $(C)$ . Find  $(D)$  as the intersection of tangents at  $(B)$  and  $(C)$ .
  - Reflection: Calculate coordinates of  $(C)$  using Law of Cosines for  $(\cos A)$ . Derive equations of tangents at  $(B)$  and  $(C)$  to find  $(D)$ .
  - Improved Solution: Parametrize line  $(AD)$  and find intersection  $(P)$  with  $(\omega)$  via circle equation. Compute  $(AP)$  as distance between  $(A)$  and  $(P)$ .

- Sub-problem 3 (Trigonometric Approach):
  - Initial Approach: Use  $(\cos A)$ ,  $(\cos B)$ ,  $(\cos C)$  to derive  $(AP)$  via angle relations.
  - Reflection: Compute  $(\cos A = \frac{11}{25})$ ,  $(\cos B = \frac{15}{13})$ ,  $(\cos C = \frac{13}{15})$ . Use Law of Sines for circumradius  $(R)$ .
  - Improved Solution: Calculate  $(AD)$  via Law of Cosines in  $(\triangle AOD)$  and use  $(\cos \angle OAD)$  to find  $(AP = 2R \cos \angle OAD)$ .

Integration:
Combine coordinate geometry and trigonometric results to confirm  $(AP = \frac{100}{13})$ .

Overall Reflection:
Both methods confirm  $(AP = \frac{100}{13})$ . The coordinate approach directly computes  $(AP)$ , while the trigonometric method uses circle properties and angle relations. Both require careful computation but align in the final result.

Final Optimization:
Verify calculations for consistency. Ensure all steps adhere to geometric principles and algebraic accuracy.

Summary:
Using coordinate geometry and trigonometric identities,  $(AP)$  is determined to be  $(\frac{100}{13})$ , leading to  $(m+n = 113)$ .
</end_of_thought|>

<|begin_of_solution|>
Solution:
1. Calculate  $(\cos A)$ :

```

963

964

965

966

967

968

969

970

971

972

973

```

\[\cos A = \frac{AB^2 + AC^2 - BC^2}{2 \cdot AB \cdot AC}
\Rightarrow AC = \frac{5^2 + 10^2 - 9^2}{2 \cdot 5}
\Rightarrow 10 = \frac{11}{25}
\]

```

2. **Coordinates of C :**

```

\[\left(8, \frac{24}{5}\right)
\Rightarrow \text{using } \cos A \text{ and } \sin A = \frac{6\sqrt{14}}{25}
\]

```

3. **Equations of tangents at B and C :**

```

- Tangent at  $B$ :  $x = 5$ 
- Tangent at  $C$ :  $y = -\frac{4}{3}x + \frac{50}{3}$ 
- Intersection  $D$ :  $\left(5, \frac{50}{3}\right)$ 

```

4. **Equation of line AD :**

```

\[\left[ y = \frac{10}{3}x \right]
\]

```

5. **Intersection P with ω :**

```

- Solve circle equation  $(x^2 + y^2 - \frac{130}{9}x - \frac{280}{27}y = 0)$  with
 $y = \frac{10}{3}x$ 
-  $P = \left(\frac{200}{39}, \frac{200}{117}\right)$ 

```

6. **Distance AP :**

```

\[\left[ AP = \sqrt{\left(\frac{200}{39}\right)^2 + \left(\frac{200}{117}\right)^2} = \frac{100}{13} \right]
\]

```

7. **Final Answer:**

```

\[\boxed{m + n = 100 + 13}
\]

```

Output from Qwen2.5-7B-Instruct(long)

<|begin_of_thought|>

<|begin_of_sentence|><|User|>Let ABC be a triangle inscribed in circle ω . Let the tangents to ω at B and C intersect at point D , and let \overline{AD} intersect ω at P . If $AB=5$, $BC=9$, and $AC=10$, AP can be written as the form $\frac{m}{n}$, where m and n are relatively prime integers. Find $m + n$.

Okay, so I have this geometry problem here about triangle ABC inscribed in a circle ω . The tangents at points B and C intersect at point D , and then line AD intersects the circle again at point P . We're given the sides $AB=5$, $BC=9$, and $AC=10$. We need to find the length of AP , express it as a reduced fraction m/n , and then find $m+n$.

First, I should probably draw a diagram to visualize the problem. Let me sketch triangle ABC with $AB=5$, $BC=9$, $AC=10$. Since the sides are $5, 9, 10$, it might not be a right-angled triangle, but let me check: $5^2 + 9^2 = 25 + 81 = 106$, which is not equal to $10^2=100$, so nope, not a right triangle. So it's a scalene triangle.

Next, the circle ω is the circumcircle of triangle ABC . The tangents at B and C meet at D . So, point D is the ex-pole of BC with respect to ω , I think. Wait, no, actually, the point where two tangents meet is called the ex-similicenter? Maybe I should recall that the point where two tangents meet is called the ex-tangent point or something. But regardless, D is the point where the two tangents from B and C meet.

Since D is the intersection of the two tangents, by the property of tangents, $DB = DC$. So triangle DBC is isosceles with $DB = DC$.

Now, we need to find point P where line AD intersects the circumcircle ω again. So, starting from A , going through D , and intersecting ω again at P . We need to find the length AP .

Hmm, okay. So, maybe I can use some power of point theorem here. The power of point D with respect to ω is equal to DB^2 (since DB is tangent to ω). Similarly, it's equal to DC^2 as well, which is consistent.

Also, since D lies outside the circle, and AD is a secant line passing through D and intersecting the circle at A and P , the power of D with respect to ω is equal to $DA \cdot DP$. So, power of D : $DB^2 = DA \cdot DP$.

So, if I can find DB and DA , I can find DP , and then subtract DA to find AP .

But wait, I don't know DB or DA yet. Hmm. Maybe I can find DB in terms of the sides of the triangle or something.

Alternatively, maybe I can use coordinate geometry or trigonometry to solve for AP . But let me see if there's a synthetic approach first.

Let me recall that in triangle ABC , if D is the intersection of the tangents at B and C , then D is the exsimilicenter of the circumcircle ω with respect to BC . Wait, maybe not exactly, but perhaps some properties can be used here.

Alternatively, since D is the intersection of the tangents at B and C , the line AD is called the symmedian. Wait, is that right? Because the symmedian is the reflection of the median over the angle bisector, but in this case, AD is connecting A to the exsimilicenter, which might be related to the symmedian.

Wait, actually, yes! I think that the point D is the exsimilicenter of the circumcircle, and AD is the symmedian. So, the line AD is the symmedian from A .

In that case, maybe I can use the properties of the symmedian. I recall that the symmedian divides the opposite side in the ratio of the squares of the adjacent sides. Wait, but in this case, since AD is the symmedian, it might intersect BC at some point, but in our problem, AD is extended to meet the circumcircle again at P .

Alternatively, maybe I can use the power of point A with respect to the circle. Wait, but A is on the circle, so its power is zero. Hmm, that might not help directly.

Wait, but D is outside the circle, so power of D is $DB^2 = DA \cdot DP$, as I thought earlier. So, if I can find DB and DA , I can find DP , and then $AP = DP - DA$.

So, let's see. First, let's compute DB . Since D is the intersection of the tangents at B and C , and in triangle ABC , the lengths of the tangents from D to ω are equal, so $DB = DC$.

But how do I compute DB? Maybe using the formula for the length of the tangent from a point to a circle. The formula is $\sqrt{(\text{distance from point to center})^2 - \text{radius}^2}$. But I don't know the radius or the coordinates of the center yet.

Alternatively, maybe I can find DB using triangle properties.

Wait, perhaps using the formula for the length of the tangent from D to ω . The tangent length squared is equal to the power of D with respect to ω , which is $DB^2 = DC^2 = OD^2 - R^2$, where O is the center of ω and R is its radius.

But since I don't know O or R, maybe another approach is better.

Alternatively, maybe I can use coordinates. Let me assign coordinate system to triangle ABC.

Let me place point A at $(0, 0)$, point B at $(5, 0)$, but wait, given the sides $AB=5, BC=9, AC=10$, that might complicate things. Alternatively, perhaps placing point B at $(0,0)$, point C at $(9,0)$, but then point A would be somewhere in the plane.

Wait, let me compute the coordinates of triangle ABC.

Given sides $AB=5, BC=9, AC=10$. So, let's place point B at $(0,0)$, point C at $(9,0)$. Then point A is somewhere in the plane. Let me compute the coordinates of A.

Let's denote coordinates: $B(0,0), C(9,0)$. Let's denote $A(x,y)$. Then, distance from A to B is 5, so $\sqrt{x^2 + y^2} = 5$, so $x^2 + y^2 = 25$.

Distance from A to C is 10, so $\sqrt{(x - 9)^2 + y^2} = 10$, so $(x - 9)^2 + y^2 = 100$.

Subtracting the first equation from the second: $(x - 9)^2 + y^2 - (x^2 + y^2) = 100 - 25 \Rightarrow x^2 - 18x + 81 + y^2 - x^2 - y^2 = 75 \Rightarrow -18x + 81 = 75 \Rightarrow -18x = -6 \Rightarrow x = 1/3$.

So, $x = 1/3$. Then, from $x^2 + y^2 = 25: (1/3)^2 + y^2 = 25, 1/9 + y^2 = 25, y^2 = 25 - 1/9 = 224/9, y = \sqrt{224}/3 = (4 * \sqrt{14})/3$.

So, coordinates of A are $(1/3, (4\sqrt{14})/3)$.

Okay, so now we have coordinates for A, B, C.

Now, we need to find point D, which is the intersection of the tangents at B and C to the circumcircle ω .

First, let me find the equation of the circumcircle ω . Since we have coordinates for A, B, C, we can find the equation of ω .

The general equation of a circle is $x^2 + y^2 + Dx + Ey + F = 0$. Plugging in points A, B, C:

For $B(0,0): 0 + 0 + 0 + 0 + F = 0 \Rightarrow F = 0$.

For $C(9,0): 81 + 0 + 9D + 0 + 0 = 0 \Rightarrow 81 + 9D = 0 \Rightarrow D = -9$.

For $A(1/3, (4\sqrt{14})/3): (1/3)^2 + ((4\sqrt{14})/3)^2 + D*(1/3) + E*(4\sqrt{14})/3 + F = 0$.

Compute each term:

$(1/3)^2 = 1/9$.

$((4\sqrt{14})/3)^2 = (16 * 14)/9 = 224/9$.

$D = -9$, so $D*(1/3) = -9*(1/3) = -3$.

$E*(4\sqrt{14})/3$ remains as is.

$F = 0$.

So, adding them up:

$$1/9 + 224/9 - 3 + (4\sqrt{14} E)/3 = 0.$$

Combine the fractions:

$$(1 + 224)/9 = 225/9 = 25.$$

$$\begin{aligned} \text{So, } 25 - 3 + (4\sqrt{14} E)/3 = 0 &\Rightarrow 22 + (4\sqrt{14} E)/3 \\ \Rightarrow 0 &\Rightarrow (4\sqrt{14} E)/3 = -22 \Rightarrow E = (-22 * \\ &\Rightarrow 3)/(4\sqrt{14}) = (-66)/(4\sqrt{14}) = \\ &\Rightarrow (-33)/(2\sqrt{14}). \end{aligned}$$

Rationalizing the denominator: $(-33\sqrt{14})/(2*14) = (-33\sqrt{14})/28$.

So, equation of ω is $x^2 + y^2 - 9x - (33\sqrt{14})/28 y = 0$.

Hmm, that seems a bit messy, but okay.

Alternatively, maybe I can compute the circumradius R using the formula:

$R = (a b c)/(4 \Delta)$, where a, b, c are sides, and Δ is the area.

Given sides $AB=5, BC=9, AC=10$.

First, compute the area Δ using Heron's formula.

Perimeter semi-perimeter $s = (5 + 9 + 10)/2 = 24/2 = 12$.

Area $\Delta = \sqrt{s(s-a)(s-b)(s-c)} = \sqrt{12*7*3*2} = \sqrt{12*7*6} = \sqrt{504} = \sqrt{16*31.5}$ = wait, 504 factors: $16*31.5$ is not helpful. Let me factor 504: $504 = 16*31.5$? Wait, 504 divided by 16 is 31.5, which is not integer. Alternatively, $504 = 9*56 = 9*7*8 = 9*7*8$. So $\sqrt{504} = \sqrt{9*7*8} = 3*\sqrt{56} = 3*\sqrt{4*14} = 3*2*\sqrt{14} = 6\sqrt{14}$.

So, area $\Delta = 6\sqrt{14}$.

Therefore, circumradius $R = (a b c)/(4 \Delta) = (5 * 9 * 10)/(4 * 6\sqrt{14}) = (450)/(24\sqrt{14}) = (75)/(4\sqrt{14}) = (75\sqrt{14})/(4*14) = (75\sqrt{14})/56$.

So, $R = (75\sqrt{14})/56$.

Okay, so the circumradius is $(75\sqrt{14})/56$.

Now, to find point D, which is the intersection of the two tangents at B and C.

The tangent at B to circle ω has the equation given by the tangent formula. For a circle with center (h, k) and radius R, the tangent at point (x_1, y_1) is $(x_1 - h)(x - h) + (y_1 - k)(y - k) = R^2$. Wait, no, that's the equation of the circle itself.

Wait, the equation of the tangent at point (x_1, y_1) on the circle is $(x_1)(x) + (y_1)(y) + D*(x + x_1)/2 + E*(y + y_1)/2 + F = 0$. Wait, maybe it's better to use the derivative or point-form.

Alternatively, since we have the equation of the circle, we can find the equation of the tangent at $B(0,0)$.

The general equation of the tangent to the circle $x^2 + y^2 + Dx + Ey + F = 0$ at point (x_1, y_1) is $x x_1 + y y_1 + D*(x + x_1)/2 + E*(y + y_1)/2 + F = 0$.

Given that, for point $B(0,0)$:

Equation of tangent at B: $x*0 + y*0 + D*(x + 0)/2 + E*(y + 0)/2 + F = 0 \Rightarrow (D x)/2 + (E y)/2 + F = 0$.

But from earlier, we have $D = -9, E = (-33\sqrt{14})/28, F = 0$.

So, the tangent at B is: $(-9/2)x + (-33\sqrt{14})/56 y \Leftrightarrow = 0$.

Wait, no: $(D x)/2 + (E y)/2 + F = 0 \Rightarrow (-9/2)x + (-33\sqrt{14})/56 y + 0 = 0$.

Simplify: multiply both sides by 56 to eliminate denominators:

$$-9/2 x * 56 + (-33\sqrt{14})/56 y * 56 = 0 \Rightarrow -9*28 x \Leftrightarrow -33\sqrt{14} y = 0 \Rightarrow -252 x - 33\sqrt{14} y = 0.$$

We can divide both sides by -3: $84x + 11\sqrt{14} y = 0$.

So, equation of tangent at B: $84x + 11\sqrt{14} y = 0$.

Similarly, find the equation of tangent at C(9,0).

Using the same formula: $x x_1 + y y_1 + D(x + x_1)/2 + E(y + y_1)/2 + F = 0$.

For point C(9,0):

$$\text{Equation: } x*9 + y*0 + D*(x + 9)/2 + E*(y + 0)/2 + F = 0 \Rightarrow 9x + (-9/2)(x + 9) + (-33\sqrt{14})/56 * y/2 + \Leftrightarrow 0 = 0.$$

Wait, hold on, let me substitute correctly.

Wait, the formula is $x x_1 + y y_1 + D*(x + x_1)/2 + E*(y + y_1)/2 + F = 0$.

So, for point C(9,0):

$$x*9 + y*0 + D*(x + 9)/2 + E*(y + 0)/2 + F = 0.$$

Substitute D = -9, E = $(-33\sqrt{14})/28$, F = 0:

$$9x + (-9/2)(x + 9) + (-33\sqrt{14})/56 * y = 0.$$

Simplify term by term:

$$9x - (9/2)x - (81/2) + (-33\sqrt{14})/56 y = 0.$$

Combine like terms:

$$(9x - (9/2)x) = (18/2 - 9/2)x = (9/2)x.$$

So, equation becomes:

$$(9/2)x - 81/2 + (-33\sqrt{14})/56 y = 0.$$

Multiply both sides by 56 to eliminate denominators:

$$(9/2)x * 56 - 81/2 * 56 + (-33\sqrt{14})y = 0 \Rightarrow 9*28 x - \Leftrightarrow 81*28 + (-33\sqrt{14})y = 0 \Rightarrow 252x - 2268 \Leftrightarrow -33\sqrt{14} y = 0.$$

Bring constants to the other side:

$$252x - 33\sqrt{14} y = 2268.$$

We can divide both sides by 3:

$$84x - 11\sqrt{14} y = 756.$$

So, equation of tangent at C is $84x - 11\sqrt{14} y = 756 \Leftrightarrow 756$.

Now, we have equations of the two tangents: at B: $84x + 11\sqrt{14} y = 0$; at C: $84x - 11\sqrt{14} y = 756$.

We need to find their intersection point D. Let's solve these two equations.

Let me write them:

$$1) 84x + 11\sqrt{14} y = 0$$

$$2) 84x - 11\sqrt{14} y = 756$$

Let me add equations 1 and 2:

$$(84x + 11\sqrt{14} y) + (84x - 11\sqrt{14} y) = 0 + 756 \Rightarrow \Leftrightarrow 168x = 756 \Rightarrow x = 756 / 168.$$

Simplify: $756 / 168$. Let's divide numerator and denominator by 12: $756/12=63$, $168/12=14$. So $63/14 \Leftrightarrow = 9/2$. So $x = 9/2$.

Now, substitute $x = 9/2$ into equation 1: $84*(9/2) + 11\sqrt{14} y = 0 \Rightarrow 84*(4.5) + 11\sqrt{14} y = 0 \Rightarrow 378 + 11\sqrt{14} y = 0 \Rightarrow 11\sqrt{14} y = -378 \Rightarrow y = \Leftrightarrow -378 / (11\sqrt{14})$.

Simplify: rationalize denominator:

$$y = (-378)/(11\sqrt{14}) * (\sqrt{14}/\sqrt{14}) = \Leftrightarrow (-378\sqrt{14})/(11*14) = (-378\sqrt{14})/154.$$

Simplify numerator and denominator: $378 / 14 = 27; \Leftrightarrow 154 / 14=11$.

So $y = (-27\sqrt{14})/11$.

So, coordinates of D are $(9/2, -27\sqrt{14}/11)$.

Alright, so D is at $(9/2, -27\sqrt{14}/11)$.

Now, we need to find point P where line AD intersects the circumcircle ω again.

We have point A at $(1/3, 4\sqrt{14}/3)$ and point D at $(9/2, -27\sqrt{14}/11)$. Let me find the parametric equation of line AD.

First, compute the direction vector from A to D:

$$\Delta x = 9/2 - 1/3 = (27/6 - 2/6) = 25/6.$$

$$\Delta y = (-27\sqrt{14}/11) - (4\sqrt{14}/3) = \Leftrightarrow (-27\sqrt{14}/11 - 44\sqrt{14}/33) = (-81\sqrt{14}/33 - \Leftrightarrow 44\sqrt{14}/33) = (-125\sqrt{14})/33.$$

So, the direction vector is $(25/6, -125\sqrt{14}/33)$. We can write parametric equations as:

$$x = 1/3 + (25/6) t$$

$$y = 4\sqrt{14}/3 + (-125\sqrt{14}/33) t$$

We need to find t such that (x, y) lies on ω , which has equation $x^2 + y^2 - 9x - (33\sqrt{14})/28 y = 0$.

So, substitute x and y into the equation:

$$[1/3 + (25/6) t]^2 + [4\sqrt{14}/3 + (-125\sqrt{14}/33) t]^2 \Leftrightarrow t^2 - 9[1/3 + (25/6) t] - (33\sqrt{14})/28 \Leftrightarrow [4\sqrt{14}/3 + (-125\sqrt{14}/33) t] = 0.$$

Wow, that's going to be a bit involved. Let me compute each term step by step.

First, compute $[1/3 + (25/6) t]^2$:

$$= (1/3)^2 + 2*(1/3)*(25/6) t + (25/6 t)^2$$

$$= 1/9 + (50/18) t + (625/36) t^2$$

Simplify:

$$1/9 + (25/9) t + (625/36) t^2.$$

Second, compute $[4\sqrt{14}/3 + (-125\sqrt{14}/33) t]^2$:

$$= (4\sqrt{14}/3)^2 + 2*(4\sqrt{14}/3)*(-125\sqrt{14}/33) t \Leftrightarrow + (-125\sqrt{14}/33 t)^2$$

Compute each term:

$$(4\sqrt{14}/3)^2 = 16*14 / 9 = 224/9.$$

$$2*(4\sqrt{14}/3)*(-125\sqrt{14}/33) t = 2*(-500*14 / 99 \Leftrightarrow) t = 2*(-7000/99) t = -14000/99 t.$$

Wait, hold on: $4\sqrt{14} * (-125\sqrt{14}) = \Leftrightarrow 4*(-125)*(\sqrt{14})^2 = 4*(-125)*14 = 4*(-1750) = \Leftrightarrow -7000$.

Then, divided by $(3 \times 33) = 99$. So, $2 \times (-7000 / 99) t =$
 $\hookrightarrow -14000/99 t$.

Third term: $(-125\sqrt{14/33} t)^2 = (125^2 \times 14)/33^2$
 $\hookrightarrow t^2 = (15625 \times 14)/1089 t^2 = 218750/1089 t^2$.

So, overall, the second squared term is:

$$224/9 - 14000/99 t + 218750/1089 t^2.$$

Third term: $-9[1/3 + (25/6) t] = -9 \times (1/3) - 9 \times (25/6) t$
 $\hookrightarrow = -3 - (225/6) t = -3 - (75/2) t$.

Fourth term: $-(33\sqrt{14})/28 [4\sqrt{14/3} +$
 $\hookrightarrow (-125\sqrt{14/33}) t]$

First, compute inside the brackets:

$$4\sqrt{14/3} + (-125\sqrt{14/33}) t = (44\sqrt{14/33}) +$$

 $\hookrightarrow (-125\sqrt{14/33}) t = (44 - 125t)\sqrt{14/33}$.

Multiply by $-(33\sqrt{14})/28$:

$$= -(33\sqrt{14})/28 * (44 - 125t)\sqrt{14/33}$$

Simplify:

$$= -(\sqrt{14} * \sqrt{14})/28 * (44 - 125t)$$

$$= -(14)/28 * (44 - 125t)$$

$$= -1/2 * (44 - 125t)$$

$$= -22 + (125/2) t.$$

So, the fourth term simplifies to $-22 + (125/2) t$.

Now, putting all terms together:

$$\text{First squared term: } 1/9 + (25/9) t + (625/36) t^2$$

$$\text{Second squared term: } 224/9 - 14000/99 t + 218750/1089$$

 $\hookrightarrow t^2$

$$\text{Third term: } -3 - (75/2) t$$

$$\text{Fourth term: } -22 + (125/2) t$$

Now, combine all these:

Sum all constants:

$$1/9 + 224/9 - 3 - 22$$

$$= (1 + 224)/9 - 25$$

$$= 225/9 - 25$$

$$= 25 - 25 = 0.$$

Sum all terms with t:

$$(25/9) t - 14000/99 t - (75/2) t + (125/2) t$$

First, convert all to a common denominator. Let's use
 $\hookrightarrow 198$.

$$25/9 = (25 \times 22)/198 = 550/198$$

$$-14000/99 = (-14000 \times 2)/198 = -28000/198$$

$$-75/2 = (-75 \times 99)/198 = (-7425)/198$$

$$125/2 = (125 \times 99)/198 = 12375/198$$

So, total t terms:

$$550/198 - 28000/198 - 7425/198 + 12375/198$$

Compute numerator:

$$550 - 28000 - 7425 + 12375 = 550 + (-28000 - 7425 + 12375)$$

 $\hookrightarrow = 550 + (-28000 - 7425 + 12375)$

Compute inside the brackets:

$$-28000 - 7425 = -35425; -35425 + 12375 = -23050$$

$$\text{So total numerator: } 550 - 23050 = -22500$$

Thus, t terms: $-22500/198 =$ Simplify: divide
 \hookrightarrow numerator and denominator by 6: $-3750/33$.

$$\text{So, } -3750/33 t.$$

Now, sum all t^2 terms:

$$625/36 t^2 + 218750/1089 t^2$$

Convert to common denominator, which is $1089 \times 4 = 4356$.

$$625/36 = (625 \times 121)/4356 = 75625/4356$$

$$218750/1089 = (218750 \times 4)/4356 = 875000/4356$$

$$\text{So total } t^2 \text{ terms: } (75625 + 875000)/4356 =$$

 $\hookrightarrow 950625/4356$.

Simplify: divide numerator and denominator by 9:

$$950625 / 9 = 105625; 4356 / 9 = 484.$$

$$\text{So, } 105625/484 t^2.$$

So, putting it all together, the equation is:

$$0 + (-3750/33) t + (105625/484) t^2 = 0.$$

Multiply both sides by 484×33 to eliminate
 \hookrightarrow denominators:

$$-3750/33 * 484 * 33 t + 105625/484 * 484 * 33 t^2 = 0 \Rightarrow$$

 $\hookrightarrow -3750 * 484 t + 105625 * 33 t^2 = 0.$

Compute coefficients:

$$-3750 * 484: \text{ Let's compute } 3750 * 484.$$

$$\text{First, } 3750 * 400 = 1,500,000$$

$$3750 * 80 = 300,000$$

$$3750 * 4 = 15,000$$

$$\text{Total: } 1,500,000 + 300,000 + 15,000 = 1,815,000$$

$$\text{So, } -3750 * 484 = -1,815,000.$$

Similarly, $105625 * 33$:

$$105625 * 30 = 3,168,750$$

$$105625 * 3 = 316,875$$

$$\text{Total: } 3,168,750 + 316,875 = 3,485,625.$$

So, the equation becomes:

$$-1,815,000 t + 3,485,625 t^2 = 0.$$

Factor t:

$$t (-1,815,000 + 3,485,625 t) = 0.$$

$$\text{Solutions are } t=0 \text{ and } t = 1,815,000 / 3,485,625.$$

$$\text{Simplify } t: 1,815,000 / 3,485,625.$$

Divide numerator and denominator by 75:

$$1,815,000 / 75 = 24,200$$

$$3,485,625 / 75 = 46,475$$

Wait, $1,815,000 / 75 = 24,200$? Wait,
 $\hookrightarrow 75 \times 24,200 = 75 \times 24,000 = 1,800,000$ and $75 \times 200 = 15,000$,
 \hookrightarrow so total $1,815,000$. Correct.

Similarly, $3,485,625 / 75 = 46,475$. Let me check:
 $\hookrightarrow 75 \times 46,475 = 75 \times (46,000 + 475) = 75 \times 46,000 + 75 \times 475 = 3,450,000 + 35,625 = 3,485,625$. So total $3,450,000 + 35,625 = 3,485,625$. Correct.

So, $t = 24,200 / 46,475$.

Simplify further: divide numerator and denominator by $\hookrightarrow 25$:

$24,200 / 25 = 968$

$46,475 / 25 = 1,859$

So, $t = 968 / 1,859$.

Check if 968 and 1,859 have a common factor. $968 / 2 = 484$; $1,859 / 2 = 929.5$, not integer. $968 / 13 = 74.46$, not integer. $1,859 / 13 = 143$, since $13 \times 143 = 1,859$. So, 1,859 = 13×143 . $968 / 13 = 74.46$, not integer. So, 968 and 1,859 are co-prime? Wait, $968 = 8 \times 121 = 8 \times 11^2$. $1,859 = 13 \times 143 = 13 \times 11 \times 13$. Wait, $143 = 11 \times 13$. So, $1,859 = 13 \times 11 \times 13 = 11 \times 13^2$.

So, $968 = 8 \times 11^2$, $1,859 = 11 \times 13^2$. So, common factor is $\hookrightarrow 11$.

So, divide numerator and denominator by 11:

$968 / 11 = 88$

$1,859 / 11 = 169$

So, $t = 88 / 169$.

Thus, $t = 88 / 169$ is the non-zero solution.

So, the point P corresponds to $t = 88 / 169$.

Therefore, coordinates of P:

$$x = 1/3 + (25/6) \times (88/169) = 1/3 + (25 \times 88) / (6 \times 169)$$

Compute $25 \times 88 = 2200$; $6 \times 169 = 1014$.

So, $x = 1/3 + 2200 / 1014$.

Simplify $2200 / 1014$: divide numerator and denominator \hookrightarrow by 2: $1100 / 507$.

So, $x = 1/3 + 1100 / 507$.

Convert $1/3$ to $169 / 507$: $1/3 = 169 / 507$.

So, $x = 169 / 507 + 1100 / 507 = 1269 / 507$.

Simplify $1269 / 3 = 423$; $507 / 3 = 169$. So, $x = 423 / 169$.

Similarly, $y = 4\sqrt{14/3} + (-125\sqrt{14/33}) \times (88/169)$.

Compute $(-125\sqrt{14/33}) \times (88/169)$:

$$= (-125 \times 88) / (33 \times 169) \sqrt{14}$$

Simplify $88/33 = 8/3$, so:

$$= (-125 \times 8) / (3 \times 169) \sqrt{14} = (-1000) / (507) \sqrt{14}$$

So, $y = 4\sqrt{14/3} - 1000\sqrt{14} / 507$.

Convert $4\sqrt{14/3}$ to $676\sqrt{14} / 507$.

So, $y = 676\sqrt{14} / 507 - 1000\sqrt{14} / 507 = (-324\sqrt{14}) / 507$.

Simplify: divide numerator and denominator by 3:

$$-108\sqrt{14} / 169$$

So, coordinates of P are $(423/169, -108\sqrt{14}/169)$.

Now, we need to compute AP.

Point A is at $(1/3, 4\sqrt{14/3})$, point P is at $\hookrightarrow (423/169, -108\sqrt{14}/169)$.

Compute distance AP.

Use distance formula:

$$AP = \sqrt{(x_P - x_A)^2 + (y_P - y_A)^2}$$

Compute $x_P - x_A$: $423/169 - 1/3 = (423 \times 3 - 169) / 507 = (1269 - 169) / 507 = 1100 / 507$.

Compute $y_P - y_A$: $(-108\sqrt{14}/169) - (4\sqrt{14/3}) = \hookrightarrow (-108\sqrt{14}/169 - 676\sqrt{14}/507) =$ convert to common denominator 507:

$$= (-324\sqrt{14}/507 - 676\sqrt{14}/507) = \hookrightarrow (-1000\sqrt{14}) / 507$$

So, $AP = \sqrt{(1100/507)^2 + (-1000\sqrt{14}/507)^2}$.

Factor out $(1/507)^2$:

$$= \sqrt{(1100^2 + (1000^2) \times 14) / 507^2} = \sqrt{(1,210,000 + 14,000,000) / 507^2} = \sqrt{15,210,000 / 507^2}$$

Compute $\sqrt{15,210,000} / 507$.

$$\sqrt{15,210,000} = \sqrt{15,210 \times 1000} = \sqrt{15,210} \times \sqrt{1000}$$

$$\hookrightarrow *1000 = \sqrt{15,210} \times \sqrt{1000}$$

But $15,210 = 1521 \times 10 = (39^2) \times 100^2 = 39^2 \times 100^2$, \hookrightarrow so $\sqrt{15,210} = 39 \times 100 = 3900$.

Wait, wait: $15,210,000 = 15,210 \times 1000 = 1521 \times 10 \times 1000 = (39^2) \times 100^2 = 39^2 \times 100^2$, $\hookrightarrow (39)^2 \times (100)^2$, so $\sqrt{15,210,000} = 39 \times 100 = 3900$.

Yes, because $39^2 = 1521$, $100^2 = 10,000$, so $\hookrightarrow \sqrt{15,210,000} = 3900$.

Thus, $AP = 3900 / 507$.

Simplify $3900 / 507$.

Divide numerator and denominator by 3: $1300 / 169$.

$1300 / 13 = 100$; $169 / 13 = 13$.

So, $1300 / 169 = 100 / 13$.

Thus, $AP = 100 / 13$.

So, $m = 100$, $n = 13$, which are coprime.

Thus, $m + n = 113$.

****Final Answer****
 $\boxed{113}$

984
985
986
987
988
989

While both models successfully reach the correct final answer ($m + n = 113$), their reasoning processes demonstrate fundamental differences in efficiency and logical clarity. We can conclude several insights below:

- 990 • **Insight 1: Logical Directness and Expert Intuition.** The Long CoT demonstrates a rigid 991 “overthinking” style characterized by mechanical trial-and-error and significant “thinking noise” (e.g., conversational self-corrections 992 like “Wait, no...” or “Hmm, maybe...”). In 993 contrast, the Short CoT model exhibits expert-like intuition by immediately identifying the 994 optimal coordinate geometry and trigonometric framework. It bypasses low-efficiency ex- 995 996 997 998 999

1000 ploratory paths, focusing directly on pivotal
1001 properties—such as the symmedian nature of
1002 the line AD —without the cognitive clutter
1003 found in the teacher-distilled baseline.

- 1004 • **Insight 2: Noise Filtering and Information**
1005 **Density.** A critical flaw in the Long CoT is its
1006 inclusion of redundant intermediate algebra,
1007 such as performing step-by-step long division
1008 for large integers. As argued in before, such
1009 verbosity can act as “noise” that interferes
1010 with the learning performance of student mod-
1011 els. The Short CoT successfully filters this
1012 noise, internalizing the mechanical calcula-
1013 tions and preserving only the “pivotal logi-
1014 cal nodes” (e.g., $\cos A$ calculation, coordinate
1015 determination of D , and circle intersection).
1016 This results in a significantly higher informa-
1017 tion density per token.

- 1018 • **Insight 3: Order-of-Magnitude Efficiency.**
1019 The quantitative difference in token consump-
1020 tion is striking. While the Long CoT con-
1021 sumes several thousand tokens to solve the
1022 problem, the Short CoT reaches the same re-
1023 sult using approximately 400 tokens. This
1024 reduction aligns with our experimental find-
1025 ings that the DAD method can compress in-
1026 ference costs by over 92.8%. This case study
1027 serves as empirical evidence that LLM models
1028 do not sacrifice accuracy for brevity; instead,
1029 they learn a more precise and essential reason-
1030 ing pathway that drastically reduces computa-
1031 tional overhead during deployment.