DA-CoTD: Efficient Chain-of-Thought Reasoning with Difficulty-Aware CoT-Distillation

Abdul Waheed* **Chancharik Mitra*** **Laurie Z. Wang***
Carnegie Mellon University
{abdulw,cmitra,zihaow3}@andrew.cmu.edu

Abstract

Chain-of-thought (CoT) prompting improves reasoning in large language models (LLMs) but often produces overly verbose traces, leading to inefficiency at inference time. This issue is amplified in multimodal reasoning, where images require greater token budget and simple problems require little reasoning while complex ones demand detailed cross-modal chains. We propose *Difficulty-Aware CoT Distillation* (DA-CoTD), a framework that adapts reasoning length to input complexity. Using an LLM-based grader aligned with AoPS difficulty ratings, we compress verbose CoT traces into difficulty-aligned ones and fine-tune multimodal models via supervised fine-tuning (SFT) and direct preference optimization (DPO). Experiments on seven multimodal math benchmarks show that DA-CoTD reduces reasoning tokens by up to 30% while maintaining or improving accuracy, outperforming strong baselines.

1 Introduction

Chain-of-thought (CoT) prompting [Wei et al., 2022] has proven effective for improving reasoning in large language models (LLMs) across domains such as multimodal reasoning tasks [Chu et al., 2024, Patil, 2025, DeepSeek-AI et al., 2025, OpenAI, 2024]. By producing intermediate steps, CoT enables complex inference, but often at the cost of excessive verbosity. This "over-thinking" increases latency, token usage, and energy costs [Chen et al., 2025b, Samsi et al., 2023].

In multimodal reasoning, inefficiency is more pronounced: some problems need only simple visual or textual cues, while others demand detailed cross-modal reasoning. Yet most CoT-enabled models apply a uniform strategy across inputs, unlike humans who adjust effort based on task complexity. Recent methods such as L1 [Aggarwal et al., 2023] and "Learning How Hard to Think" [Damani et al., 2024] attempt to control reasoning length or adapt compute dynamically. However, they rely on fixed budgets or external routing, rather than teaching models to internally modulate reasoning depth.

To address this problem, we propose *DA-CoTD*, where models adapt reasoning length to input difficulty. Using an LLM-based grader based on AoPS[2] ratings, we estimate problem difficulties and compress verbose CoT traces into difficulty-aligned ones. These traces are used to fine-tune models with supervised fine-tuning (SFT) and direct preference optimization (DPO) [Rafailov et al., 2024].

Our experiments address two questions: **(RQ1)** Can difficulty-adaptive models match or exceed the accuracy of full CoT models while reducing tokens? **(RQ2)** Do they provide better efficiency–accuracy trade-offs than static or length-controlled baselines? Results show up to 30% token reduction with minimal or no accuracy loss, demonstrating that difficulty-aware multimodal reasoning enables efficient, interpretable, and adaptive inference.

*Equal contribution. Laurie's debugging efforts were generously supported by her avian companion (a bird), whose contributions we also warmly acknowledge.

[2] https://artofproblemsolving.com/wiki/index.php/AoPS_Wiki:Competition_ratings

2 Related Work

Chain-of-Thought Reasoning. Chain-of-Thought (CoT) prompting [Wei et al., 2022] improves reasoning in LLMs by producing intermediate steps, with extensions to zero-shot [Kojima et al., 2022], adaptive [Reid et al., 2023], complexity-based [Fu et al., 2022], and structured variants such as self-consistency [Wang et al., 2022], Tree-of-Thought [Yao et al., 2023], and Graph-of-Thought [Besta et al., 2023, Hao et al., 2023] further boost performance by enabling diverse or compositional reasoning. In multimodal domains, CoT has been adapted for joint visual–textual reasoning [Zhang et al., 2023, Zheng et al., 2023, Pi et al., 2023], improving accuracy on visual question answering and medical reasoning tasks [Liu et al., 2023, Lu et al., 2022, Wu et al., 2023].

Efficient and Length-Controlled Reasoning. Prior works in efficient reasoning like L1 [Li et al., 2024] use reinforcement learning for concise rationales, while verifier-guided distillation [Zhou et al., 2024] and prompt compression [Guo et al., 2024, Wingate et al., 2022] reduce redundancy. Multimodal methods like SCOTT [Wang et al., 2023] and VLAA-Thinker [Chen et al., 2025a] apply staged summarization to cut token usage. Other strategies include latent reasoning [Lang et al., 2024], rationale distillation [Hsieh et al., 2023, Muennighoff et al., 2025], activation/context compression [Zhang et al., 2024a, Ge et al., 2024, Mu et al., 2024, Chevalier et al., 2023], and inference acceleration via speculative or multi-head decoding [Leviathan et al., 2023, Miao et al., 2024, Cai et al., 2024].

3 Method

We propose a framework for difficulty-aware chain-of-thought (CoT) distillation that adapts reasoning verbosity to problem difficulty. The method has two stages: (1) generating difficulty-aligned data by compressing verbose reasoning traces, and (2) training models with supervised fine-tuning (SFT) and direct preference optimization (DPO). The central idea is to teach models to imitate traces matched to input complexity, enabling efficient and interpretable reasoning without sacrificing accuracy.

3.1 Data Generation

Given a set of math problems $\{x_i\}_{i=1}^N$ with long teacher-generated traces r_i^{long}, we compress these traces into concise versions aligned with task difficulty.

Difficulty Estimation. We grade each problem on a scalar difficulty score $d(x_i) \in [1, 10]$, estimated using `GPT-4o-mini` (temperature=0). The scoring prompt aligns with Art of Problem Solving (AoPS) ratings, following SkyThought [Team, 2025]. This provides stable, human-aligned estimates of reasoning complexity.

Difficulty-Aware Compression. Conditioned on $d(x_i)$, each long trace r_i^{long} is compressed into $\tilde{r}_i$ via few-shot prompting:

$$\tilde{r}_i = s(r_i^{\text{long}}, d(x_i)),$$

where $s(\cdot, \cdot)$ adaptively adjusts verbosity—short traces for easy problems, detailed chains for hard ones—while preserving logical correctness and final answers. The resulting dataset consists of tuples $(x_i, \tilde{r}_i, y_i)$, where y_i is the ground-truth solution.

In our analysis, we find that easier problems (difficulty ≤ 5) can be compressed by about 50%, whereas harder problems achieve around 30% compression on average.

Data. We use subset of LLaVA-CoT-100K [Xu et al., 2024a], comprising approximately 60,000 examples. Each example includes an extended chain-of-thought generated by **GPT-4o**, which we then condense using **GPT-4o-mini** to produce concise summaries. These summarized reasoning trajectories serve as the training data for our multimodal models.

3.2 Training

After generating difficulty-aligned data, we fine-tune student models to learn adaptive reasoning behavior, using Qwen2.5-VL (3B and 7B-Instruct) as the base models.

Supervised Fine-Tuning (SFT). SFT is the first stage, where the model learns to map each problem to its compressed trace:

$$\min_{\theta} \ \mathcal{L}_{\text{SFT}} = \sum_{i=1}^{N} \text{CE}\big(f_\theta(x_i), \tilde{r}_i\big),$$

with CE denoting cross-entropy loss. This step teaches the model to follow reasoning styles aligned with input difficulty.

Direct Preference Optimization (DPO). Next, we refine reasoning with DPO [Rafailov et al., 2024]. For each input, the compressed trace $\tilde{r}_i$ is marked as preferred and the original verbose trace r_i^{long} as rejected:

$$\min_\theta \ \mathcal{L}_{\text{DPO}} = -\sum_{i=1}^{N} \log \frac{\exp\Big(\beta \cdot \text{KL}(f_\theta(x_i), \tilde{r}_i)\Big)}{\exp\Big(\beta \cdot \text{KL}(f_\theta(x_i), \tilde{r}_i)\Big) + \exp\Big(\beta \cdot \text{KL}(f_\theta(x_i), r_i^{\text{long}})\Big)},$$

where β is a temperature and KL denotes divergence. This objective nudges the model to prefer concise reasoning.

Hybrid Training. Finally, we combine SFT and DPO: first imitation, then refinement. This two-stage setup helps models acquire difficulty-aware reasoning patterns and then improve them through preference optimization.

4 Experimental Setup

We evaluate our difficulty-aware CoT distillation (DA-CoTD) in the multimodal setting, comparing against strong baselines across diverse benchmarks.

Baseline Models. We compare against two categories of baselines: (1) **Large fine-tuned models**, such as LLaVA-CoT-11B [Xu et al., 2024b], representing strong multimodal reasoning; and (2) **Base models in zero-shot settings**, including Qwen2.5-VL-3B and Qwen2.5-VL-7B-Instruct. These baselines provide reference points for both efficiency and accuracy.

Training Data. Multimodal training data is created from LLaVA-CoT [Xu et al., 2025], totaling about 60K examples. Due to resource limits, we filter this to 6K randomly selected samples. For SFT, we use compressed CoT traces as ground truth; for DPO, compressed traces serve as positive examples and verbose traces as negatives.

Evaluation. We evaluate on seven multimodal reasoning benchmarks—**MathVista** [Lu et al., 2024], **MathVerse** [Zhang et al., 2024b], **HLE (V)** [Phan et al., 2025], **MathVision** [Wang et al., 2024], **OlympiadBench (V)** [He et al., 2024], **MMStar** [Chen et al., 2024], and **MMMUPro** [Yue et al., 2024]. We restrict to single-image, free-form tasks and measure *pass@1* accuracy and average reasoning token usage, verifying answers with `GPT-4o-mini` (temperature=0) when outputs deviate from expected formats.

Implementation Details. All models are trained for three epochs with a maximum sequence length of 4096 using LLaMA-Factory [Zheng et al., 2024]. Training is performed on 8×H100 GPUs with mixed precision (fp16) and greedy decoding, and evaluation is conducted with `VLMEvalKit` [Duan et al., 2024]. For computational feasibility, we adopt the default hyperparameters from LLaMA-Factory unless otherwise specified.

All hyperparameters are provided in Appendix A.1, the prompts in Appendix A.2, and additional results and analysis in Appendix A.4.

5 Results

In our main results, we report *pass@1* with a verifier and average token counts across multimodal benchmarks in Figure 1. We observe that our SFT student matches or outperforms smaller baselines on several tasks: e.g., 28% on MathVerse (vs. 22/25 for LLaVA-CoT-11B and Qwen2.5-VL-3B)

and 51% on MMStar (vs. 45/43). DPO alone boosts OlympiadBench performance (16%, tying Qwen2.5-VL-7B) but lags on MathVision (9% vs. 14). The combined SFT+DPO model achieves the strongest overall results, e.g., 18% on MathVision (vs. 14 for Qwen2.5-VL-7B) and 51% on MMStar.

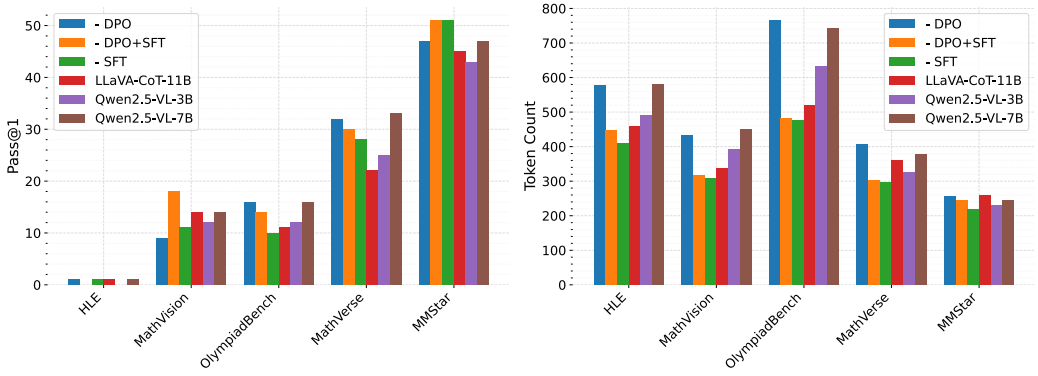


Figure 1: The left bar plot shows pass@1 and right plot shows average token count in reasoning output by different models across different multimodal benchmark. SFT, DPO, and SFT + DPO are our models trained on Qwen-2.5-VL-7B-Instruct base.

From Figure 1 (right), SFT reduces tokens by 25–30% compared to its 7B teacher—for instance, HLE (580→411) and MathVerse (378→296)—while maintaining accuracy. DPO alone generally increases token usage. The combined SFT+DPO preserves SFT's efficiency (e.g., 317 tokens on MathVision vs. 308 for SFT) while delivering stronger *pass@1* rates.

SFT vs DPO vs SFT + DPO Across tasks, SFT is most effective at reducing reasoning length while aligning with problem difficulty. DPO improves reasoning accuracy by better capturing the teacher's behavior but tends to increase verbosity. The combined SFT+DPO balances both, delivering consistent gains while preserving most of SFT's efficiency. This highlights the complementary roles of imitation (SFT) and preference alignment (DPO) in difficulty-aware distillation.

Error Analysis On multimodal benchmarks, SFT+DPO improves accuracy slightly over the base model (25.4% vs. 24.9%), with uneven gains. It shows the largest boost on MMSTAR (+4 pts) and is the only system to solve HLE-MATH-VISION, while the base remains competitive on OLYMPIADBENCH and MATHVISION.

Qualitative analysis reveals two main error types reduced by DPO: (1) formatting violations (e.g., extra rationale when only a boxed answer is expected), and (2) shallow visual grounding, such as selecting salient objects instead of reasoning over relationships. SFT+DPO mitigates these by producing cleaner outputs and more careful visual reasoning. Remaining errors—algebraic slips, ratio confusions, and hallucinations—point to the need for richer step-level supervision and more grounded multimodal training data.

6 Conclusion

We present Difficulty-Aware Chain-of-Thought Distillation (DA-CoTD), a framework that adapts reasoning verbosity to problem complexity. On multimodal benchmarks, SFT reduces token usage, DPO strengthens accuracy, and their combination balances both, turning base models like Qwen2.5-VL-7B into efficient reasoning systems. Our results show that difficulty-aware reasoning enables models to think "just enough" for each task—making them more efficient, accurate, and adaptive.

Our study has several limitations. First, the training data size was modest (6K examples) due to compute constraints. Second, our evaluation centered on mathematical reasoning, leaving other domains (e.g., spatial or compositional reasoning) for future work. Third, difficulty estimation relied on AoPS ratings, which may not generalize across tasks. Finally, experiments were limited to smaller model sizes (3B and 7B); scaling behavior across larger models remains to be tested. We encourage future works to explore further along these directions.

References

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16624–16648, 2023.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*, 2023.

Tianle Cai, Yuhong Li, Zhengmi Chen, Quoc V. Le, J. Zico Kolter Yang, and Chunyuan Li. Medusa: Simple framework for accelerating llm generation with multiple decoding heads. 2024.

Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. Sft or rl? an early investigation into training r1-like reasoning large vision-language models, 2025a. URL https://arxiv.org/abs/2504.11468.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models?, 2024. URL https://arxiv.org/abs/2403.20330.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do not think that much for 2+3=? on the overthinking of o1-like llms, 2025b. URL https://arxiv.org/abs/2412.21187.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3829–3846, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.232. URL https://aclanthology.org/2023.emnlp-main.232/.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. Navigate through enigmatic labyrinth a survey of chain of thought reasoning: Advances, frontiers and future. In *The 62nd Annual Meeting of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, August 11–16, 2024*. Association for Computational Linguistics, 2024. URL https://arxiv.org/abs/2309.15402.

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. Learning how hard to think: Input-adaptive allocation of lm computation, 2024. URL https://arxiv.org/abs/2410.04707.

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

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 11198–11201, 2024.

Yao Fu, Hao Peng, Tushar Khot, Oyvind Tafjord, Peter Clark, and Niket Tandon. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*, 2022.

Tao Ge, Heming Jing, Lei Wang, Xin Wang, Shangguang Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *International Conference on Learning Representations*, 2024.

Jiaxin Guo, Di Li, Lemao Liu, Pei Zhou, and Shuhan Liao. Softcot: Soft chain-of-thought for efficient reasoning with llms. *arXiv preprint arXiv:2402.10200*, 2024.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daiyi Peng, Zhijian Duan, Hao Sun, Kevin P. Murphy, Tze Leung Lai, Amy Wang, and Lesheng Wang. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems, 2024. URL https://arxiv.org/abs/2402.14008.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomás Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023. URL https://arxiv.org/abs/2305.02301.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35: 22199–22213, 2022.

Yang Lang, Zihao Ji, Hao Wang, Wenlin Yao, Jiacheng Xu, Ming Yan, Tao Gui, and Qi Zhang. Coconut: Combining implicit and explicit knowledge for chain-of-thought reasoning. 2024.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. 2023.

Huajian Li, Zhao Chen, J. Zico Kolter, Aditya Parameswaran, Graham Neubig, and Yizheng Alan Jiang. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2024.

Zhiyuan Liu, Yaohui Chen, Yuan Lin, Max Tegmark, and Bolei Zhou. Visual chain of thought: Bridging logical gaps with multimodal infillings. *arXiv preprint arXiv:2305.09666*, 2023.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2522, 2022.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts, 2024. URL https://arxiv.org/abs/2310.02255.

Chengcheng Miao, Mengyu Ye, Feng Wang, and Furong Huang. Seed: Accelerating reasoning tree construction via scheduled speculative decoding. 2024.

Jesse Mu, Xiang Lisa Li, and Noah D. Goodman. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36, 2024.

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

OpenAI. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Avinash Patil. Advancing reasoning in large language models: Promising methods and approaches, 2025. URL https://arxiv.org/abs/2502.03671.

Long Phan, Alice Gatti, and Others. Humanity's last exam, 2025. URL https://arxiv.org/abs/2501.14249.

Xingyu Pi, Fuxiao Wang, Shuhuai Zhang, Chenghao Fu, Chuang Fu, Eric Lo, and Yangfeng Zou. Visual chain-of-thought: Bridging logical gaps with multimodal infillings. *arXiv preprint arXiv:2305.02317*, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Machel Reid, Denny Zhou, Yao Fu, Pengcheng Liu, and Graham Neubig. Automatic prompt optimization with "gradient descent" and beam search. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023. URL https://aclanthology.org/2023.findings-emnlp.1001.

Siddharth Samsi, Dan Zhao, Joseph McDonald, Baolin Li, Adam Michaleas, Michael Jones, William Bergeron, Jeremy Kepner, Devesh Tiwari, and Vijay Gadepally. From words to watts: Benchmarking the energy costs of large language model inference. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–9. IEEE, 2023.

NovaSky Team. Sky-t1: Train your own o1 preview model within $450. https://novasky-ai.github.io/posts/sky-t1, 2025. Accessed: 2025-01-09.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=QWTCcxMpPA.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiangpeng Wei. Scott: Self-consistent chain-of-thought distillation. 2023.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

David Wingate, Mohammad Shoeybi, and Taylor Sorensen. Prompt compression and contrastive conditioning for controllability and toxicity reduction in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5621–5634, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.412. URL https://aclanthology.org/2022.findings-emnlp.412/.

Karan Wu, Jiang Jimmy Cao, Tao He, Rahul Pudipeddi, Pratik Ghanathe, Preyesh Tantia, Vikas Thaker, Yin Liu, Arsha Nagrani, Sachit Kejriwal, et al. Med-palm 2: Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617*, 2023.

Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step, 2024a. URL https://arxiv.org/abs/2411.10440.

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2024b.

Guowei Xu, Zhutian Chang, Keqiang Wu, Mingyu Xie, Benfeng Ma, Guohai Yang, and Min Wang. Llava-cot: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffith, Yangfeng Xu, and Xiong Shen. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark, 2024. URL https://arxiv.org/abs/2409.02813.

Peng Zhang, Zihan Liu, Shitao Xiao, Ningyu Shao, Qinghao Ye, and Zhicheng Dou. Soaring from 4k to 400k: Extending llm's context with activation beacon. *arXiv preprint arXiv:2401.03462*, 2024a.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems?, 2024b. URL https://arxiv.org/abs/2403.14624.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. *arXiv preprint arXiv:2310.16436*, 2023.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL `http://arxiv.org/abs/2403.13372`.

Ping Yu Zhou, Bill Yuchen Lin, Michihiro Yasunaga, and Xiang Ren. Distilling multi-step reasoning capabilities of large language models into smaller models via verifier-guided sequence generation. 2024.

A Appendix

A.1 Hyperparameters

In Table 1 we provide a detailed list of parameters used in our experiments.

Training Phase	R1-Distill-Qwen-1.5B Qwen2.5-VL-3B	R1-Distill-Qwen-7B Qwen2.5-VL-7B-Instruct
Supervised Fine-Tuning (SFT)		
LoRA rank	16	16
LoRA target	all	all
Freeze vision tower	True	True
Freeze multi-modal projector	True	True
Cutoff length	4096	4096
Per device batch size	2	2
Gradient accumulation steps	8	8
Effective batch size	128	128
Learning rate	3.0e-4	2.0e-4
Training epochs	3.0	3.0
Direct Preference Optimization (DPO)		
LoRA rank	16	16
LoRA target	all	all
Pref beta	0.1	0.1
Pref loss	sigmoid	sigmoid
Freeze vision tower	True	True
Freeze multi-modal projector	True	True
Train MM proj only	False	False
Cutoff length	4096	4096
Per device batch size	1	1
Gradient accumulation steps	8	8
Effective batch size	64	64
Learning rate	7.0e-6	5.0e-6
Training epochs	3.0	3.0
LR scheduler type	cosine	cosine
Warmup ratio	0.05	0.05

Table 1: Hyperparameters used for model training

A.2 Prompts

In this section, we present the complete set of prompts used across the various tasks in our study.

A.2.1 Rating

The prompt that we used to estimate difficulty is shown in Figure 2.

You will be given a multimodal math problem, which includes mathematical expressions and/or visual elements (such as graphs, diagrams, or charts). Your task is to grade the difficulty level from 1-10 according to the AoPS standard.

Here is the standard:
All levels are estimated and refer to averages. The following is a rough standard based on the USA tier system AMC 8 - AMC 10 - AMC 12 - AIME - USAMO/USAJMO - IMO, representing Middle School - Junior High - High School - Challenging High School - Olympiad levels. Other contests can be interpolated against this.

Notes:
- Multiple choice tests like AMC are rated as though they are free-response. Test-takers can use answer choices as hints.
- Some Olympiads are taken in 2 sessions, with similarly difficult sets of questions numbered as one sequence.

Scale
1: Strictly beginner (MOEMS, AMC 8 1-20, standard middle school problems).
2: Motivated beginners (AMC 8 21-25, AMC 10 11-20, complex word problems).
3: Creative thinking required (AMC 10 21-25, AIME 4-6).
4: Intermediate (AMC 12 21-25, AIME 7-9).
5: Difficult AIME (10-12), simple Olympiad proofs.
6: High AIME (13-15), introductory Olympiad.
7: Technical Olympiad questions (USAJMO 3/6, medium IMO 2/5).
8: Advanced Olympiad (hard IMO 2/5, easiest IMO 3/6).
9: Expert Olympiad (average IMO 3/6).
10: Extremely tedious/difficult (beyond standard competitions).

Examples
<1: Counting edges/corners/faces of a cube (2003 AMC 8 Problem 1)>.
1: Integer solutions to $|x| < 3\pi$ (2021 AMC 10B Problem 1).
2: Die roll probability with even number conditions (2021 AMC 10B Problem 18).
3: Triangle area with midpoints and angle bisectors (2018 AMC 10A Problem 24).
4: Recursive sequence interval analysis (2019 AMC 10B Problem 24).
5: System with reciprocal equations (JBMO 2020/1).
6: Acute triangle geometry with circumcircles (2020 AIME I Problem 15).
7: Balanced set existence proof (IMO 2015 Problem 1).
8: Coin collection partitioning proof (IMO 2014 Problem 5).
9: Prime circle arrangement uniqueness (IMO 2022 Problem 3).
10: Point separation line existence proof (IMO 2020 Problem 6).

Task:
Problem to be labeled: {problem}.
Consider both mathematical content and visual elements. Place difficulty in [[level]].

Important:
- Only output the numerical rating in [[ ]].
- Account for answer choice hints in multiple-choice problems.

Output: [[insert your difficulty level here]]

Figure 2: Multimodal math problem difficulty grading prompt. Extends AoPS standards to problems combining mathematical expressions with visual elements, with explicit guidance for handling multiple-choice hints and graphical components.

A.2.2 Chain-of-Thought Compression

The prompt we used to compress long chain-of-thought reasoning is shown in Figure 3.

You are given a detailed multimodal math problem solution, where the full chain-of-thought (CoT) reasoning is provided. Your task is to refine this reasoning to improve clarity, conciseness, and logical flow, while preserving all essential mathematical steps and the original problem-solving structure.
Your objective is to: - Eliminate redundancy, verbosity, and non-essential commentary - Remove unnecessary explanations or filler that do not contribute to solving the problem - Preserve all mathematically necessary steps, notation, and logical transitions - Streamline the reasoning while respecting the original depth and complexity
Instructions: The refinement should be proportional to the problem's difficulty (a higher difficulty rating indicates a harder problem). If the original solution is long and overly detailed, focus on removing only redundant or verbose elements. Do not oversimplify or compress steps aggressively — ensure the solution remains complete, accurate, and easy to follow. The difficulty rating is provided on a scale from 1 to 10, where 1 is the easiest and 10 is the hardest.
Task Guidelines: - Retain all mathematically necessary steps, logical transitions, and essential details required to understand and solve the problem correctly - Only restructure or rephrase if it clearly improves clarity without altering the logic or meaning - Eliminate redundancy, verbose phrasing, filler commentary, or repeated logic that does not contribute directly to problem-solving - Maintain mathematical accuracy, coherence, and the original formatting style - The degree of refinement (i.e., the length and detail of the refined CoT) should be proportional to the problem's difficulty — simpler problems should be more concise, while complex ones may require more detailed reasoning to arrive at the solution
Output Format: 1. Enclose the refined reasoning inside <think> and </think> tags 2. On a **new line** after </think>, write the final answer using $\boxed{final\_answer}$
Input: Problem Difficulty: {rating} Solution: {solution}
Output: <think> {{refined CoT reasoning}} </think> $\boxed{\{final\_answer\}}$

Figure 3: Multimodal chain-of-thought refinement prompt. Maintains identical structure to mathematical version while implicitly handling visual elements through preserved problem-solving structure.

A.2.3 Verification

The prompt used to verify the multimodal answers is shown in Figure 4.

You are given a question, the correct answer, and a candidate solution. Your task is to verify if the candidate solution is correct by comparing the final answer in the candidate solution with the correct answer.
Question: {question}
Correct Answer: {answer}
Solution: {solution}
Instruction: Verify whether the final answer in the candidate solution matches the correct answer. Provide your response in the format $\boxed{\text{correct}}$ if the candidate solution is correct, or $\boxed{\text{incorrect}}$ if it is incorrect. *Do not provide any additional information*.
Final Answer: $\boxed{\text{correct}}$ or $\boxed{\text{incorrect}}$

Figure 4: Answer verification prompt. Directly replicates original instructions for strict answer matching without modifications.

A.2.4 Evaluation

Figure 5 shows the base prompt used to evaluate all multimodal models.

Please answer the question below, explaining your reasoning step by step before providing the final answer.
Question: {question}

Figure 5: Step-by-step explanation prompt. Directly replicates original instructions for detailed reasoning before final answer.

A.3 LLM-as-Judge Evaluation

We provide our prompts for clarity, completeness, correction, and redundancy in Figures 6, 7, 8, 9.

You will be given a solution for a math problem. Your task is to evaluate the solution for its clarity.

Evaluation Criteria:
Clarity (1–5): Evaluates how clearly the solution communicates its reasoning.

- **Score: 5** – Exceptionally clear with perfect logical flow and insightful explanations.
- **Score: 4** – Clear and well-structured with good explanations.
- **Score: 3** – Somewhat clear but could benefit from better organization.
- **Score: 2** – Poorly structured with unclear connections between steps.
- **Score: 1** – Confusing and disorganized, very difficult to follow.

Evaluation Steps:

1. Carefully read both the problem and the provided solution.
2. Examine the logical progression from one step to the next.
3. Determine whether key concepts and reasoning steps are explained in a clear and concise manner.
4. Based on the criteria above, assign a clarity score from 1 to 5.

Instructions:

- **Do not** attempt to solve, fix, or fact-check the solution.
- Focus solely on how clearly the reasoning is communicated.
- Provide your score in the following format: `[[score]]`

Input:
Problem: {question}
Solution: {solution}

Output (score only):
`- Clarity: [[score]]`

Figure 6: Prompt for evaluating clarity of mathematical solutions.

A.4 Results

In this section, we present the additional results to support our work.

A.5 Evaluation of Summarization Approaches for CoT Compression

To develop an effective difficulty-aware reasoning framework, we first investigated different summarization techniques to compress reasoning trajectories. Our objective was to identify methods that maximize compression while preserving essential reasoning information across problems of varying difficulty.

You will be given a solution for a math problem. Your task is to evaluate the solution for its completeness.

Evaluation Criteria:
Completeness (1–5): Evaluates whether all essential reasoning steps are included.

- **Score: 5** – Perfectly complete, including all necessary steps while maintaining conciseness.
- **Score: 4** – Contains all essential reasoning steps with only minor details omitted.
- **Score: 3** – Most essential steps are included, but some connections require inference.
- **Score: 2** – Several important steps are missing, creating significant gaps.
- **Score: 1** – Critical reasoning steps are missing, making it impossible to follow.

Evaluation Steps:

1. Carefully read both the problem and the provided solution.
2. Identify the key logical and mathematical steps required to solve the problem.
3. Assess whether these steps are present and clearly stated in the solution.
4. Based on the criteria above, assign a completeness score from 1 to 5.

Instructions:

- **Do not** attempt to solve, fix, or fact-check the solution.
- Focus solely on whether all essential reasoning steps are present.
- Provide your score in the following format: `[[score]]`

Input:
Problem: {question}
Solution: {solution}

Output (score only):
`- Completeness: [[score]]`

Figure 7: Prompt for evaluating completeness of mathematical solutions.

We compared six distinct approaches to generate and summarize chains of thought. As shown in Table 2, these approaches yielded significantly different compression ratios when applied to the same set of mathematical problems.

Our baseline `long_cot` consists of full reasoning traces generated by DeepSeek-R1, representing the verbose reasoning typically produced by advanced language models. For comparison, we evaluated a single-step approach (`succinct_cot`) where we directly instructed GPT-4o to produce concise reasoning.

We then explored various two-step summarization approaches using GPT-4o to compress the original DeepSeek-R1 reasoning:

- The `gpt4o_basic` approach applied a general summarization prompt without difficulty awareness, achieving the highest compression ratio at 79.1% (reducing from 4702.3 to 982.8 tokens on average).
- The `gpt4o_num` variant enforced a structured output format with numbered reasoning steps, resulting in a 55.2% reduction.
- Finally, we tested two difficulty-aware variants: `gpt4o_DA1` incorporated difficulty ratings to guide compression level, while `gpt4o_DA2` combined difficulty awareness with the structured output format.

The difficulty-aware approach with structured output format (`gpt4o_DA2`) achieved an optimal balance, reducing token usage by 56.7% while maintaining appropriate reasoning detail proportional to problem complexity. This aligns with our goal of teaching models to "think hard" on difficult problems while being concise on simpler ones.

You will be given a solution for a math problem. Your task is to evaluate the solution for the correctness of its intermediate steps.

Evaluation Criteria:
Correctness of Intermediate Steps (1–5): Evaluates the mathematical accuracy of each intermediate reasoning step, not just the final answer.

- **Score: 5** – All intermediate reasoning steps are mathematically correct and logically sound.
- **Score: 4** – Nearly all steps are mathematically correct with only very minor errors.
- **Score: 3** – The majority of steps are mathematically sound, but there are a few minor errors.
- **Score: 2** – Some steps are mathematically incorrect or invalid logical connections.
- **Score: 1** – Most intermediate steps contain significant mathematical errors or invalid logical connections.

Evaluation Steps:

1. Carefully read both the problem and the provided solution.
2. Identify and examine each intermediate reasoning step.
3. Determine whether each step is mathematically correct and logically valid.
4. Based on the criteria above, assign a correctness score from 1 to 5.

Instructions:

- **Do not** solve or complete the problem yourself.
- Focus only on evaluating the accuracy of the intermediate steps shown in the solution.
- Provide your score in the following format: `[[score]]`

Input:
Problem: {question}
Solution: {solution}

Output (score only):
`- Correctness of Intermediate Steps: [[score]]`

Figure 8: Prompt for evaluating correctness of intermediate steps in mathematical solutions.

Setup	Avg Tokens	Max Reduction
long_cot	4702.3	0.0%
succinct_cot	3456.2	26.5%
gpt4o_basic	**982.8**	**79.1%**
gpt4o_num	2106.6	55.2%
gpt4o_DA1	2473.4	47.4%
gpt4o_DA2	2036.1	56.7%

Table 2: Summarization results and compression ratio with different prompting approaches.

While `gpt4o_basic` achieved the highest compression, our qualitative analysis revealed that it often omitted critical reasoning steps needed for more difficult problems. In contrast, `gpt4o_DA2` produced summaries that appropriately scaled with problem difficulty—remaining concise for simpler problems while preserving necessary detail for complex ones. Based on these findings, we selected `gpt4o_DA2` as our primary summarization approach for generating training data in subsequent experiments.

A.6 Error Analysis

In this section, we provide chain-of-thought reasoning traces generated by different models. In Table 3, we show a few qualitative examples of the chain-of-thought reasoning process (annotated) by our model (7B SFT + DPO) and LLava-CoT baseline.

You will be given a solution for a math problem. Your task is to evaluate the solution for its redundancy.

Evaluation Criteria:
Redundancy (1–5): Evaluates the presence of unnecessary repetition or redundant steps in the solution.

- **Score: 5** – Contains no redundancy – each step builds on previous ones without unnecessary repetition.
- **Score: 4** – Mostly free of redundancy, with minimal unnecessary repetition.
- **Score: 3** – Contains some redundancy but most repetition serves a purpose.
- **Score: 2** – Shows significant redundancy with several instances of unnecessary repetition.
- **Score: 1** – Contains extensive redundancy with multiple unnecessary repetitions.

Evaluation Steps:

1. Carefully read both the problem and the provided solution.
2. Identify any repetition or restatement of the same concepts or steps.
3. Assess whether the repetition is necessary for clarity or if it is excessive and avoidable.
4. Based on the criteria above, assign a redundancy score from 1 to 5.

Instructions:

- **Do not** attempt to solve or correct the solution.
- Focus only on identifying and evaluating unnecessary repetition or redundant reasoning.
- Provide your score in the following format: `[[score]]`

Input:
Problem: {question}
Solution: {solution}

Output (score only):
`- Redundancy: [[score]]`

Figure 9: Prompt for evaluating redundancy in mathematical solutions.

Instructions:

- **Do not** attempt to solve, fix, or fact-check the solution.
- Focus only on evaluating how clearly the reasoning is communicated.
- First, provide your reasoning inside `<think>...</think>`.
- Then, output the final clarity score in the format `<score>[[score]]</score>`.

Output:

```
<think>
[Your detailed step-by-step reasoning about the clarity of the
solution and justification for the score goes here.]
</think>

<score>[[score]]</score>
```

Figure 10: Extended clarity evaluation prompt with rationale and explicit XML-style output format.

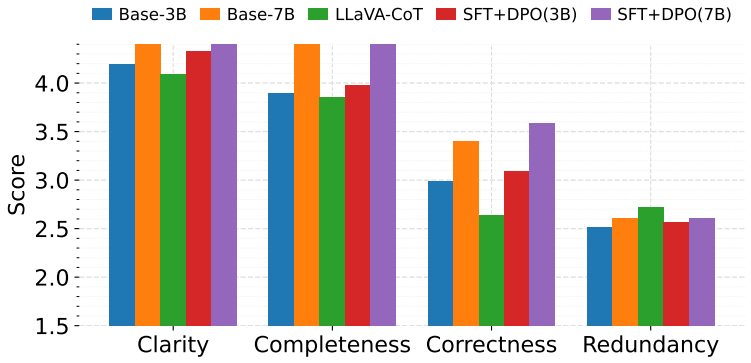


Figure 11: LLM-as-Judge evaluation scores with chain-of-thought rationale.

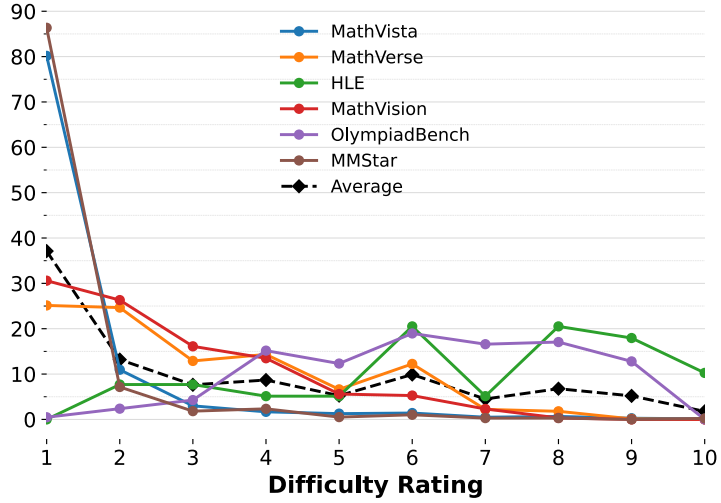


Figure 12: Difficulty ratings assigned to each problem using the AoPS scale. Multimodal evaluation datasets tend to contain easier problems overall. These ratings guide our compression process, enabling adaptive reasoning trace lengths based on problem complexity.

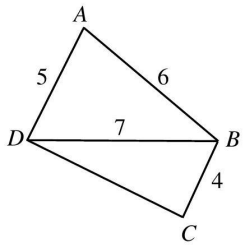
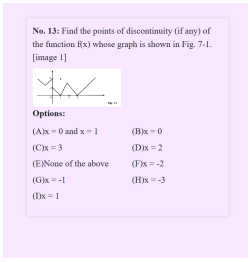
Example	Qwen-2.5-VL-7B (SFT + DPO)	LLava-CoT
 *In the diagram, $ABCD$ is a quadrilateral in which $\angle A + \angle C = 180^\circ$. What is the length of CD ?*	To find CD, we are given: $AB = 6$, $AD = 5$, $BC = 4$, $BD = 7$, and $\angle A + \angle C = 180^\circ$. Since opposite angles are supplementary, quadrilateral $ABCD$ is cyclic. **Triangle ABD:** Apply the Law of Cosines: $49 = 36+25-60\cos(\angle A) \Rightarrow \cos(\angle A) = \frac{1}{5}$ **Triangle BCD:** Using $\cos(\angle C) = -\cos(\angle A)$, apply the Law of Cosines: $49 = 16+x^2+\frac{8x}{5} \Rightarrow 5x^2+8x-165 = 0$ Solving the quadratic: $x = \frac{-8 \pm \sqrt{3364}}{10} = \frac{-8 \pm 58}{10} \Rightarrow x = 5$ **Final Answer:** $\boxed{5}$	Given $AB = 6$, $AD = 5$, $BC = 4$, and $\angle A+\angle C = 180^\circ$, the reasoning incorrectly assumes that $ABCD$ is a parallelogram. It then wrongly applies the property that opposite sides are equal: $CD = AB = 6$ **Final Answer:** $\boxed{6}$ (Incorrect)
	To determine the discontinuities of $f(x)$, we analyze the graph step-by-step: • **At $x = 0$**: Left-hand limit = 2, right-hand limit = 1. Since these limits differ, there is a **jump discontinuity**. • **At $x = 1$**: Left-hand limit = 1, right-hand limit = 2. Again, a **jump discontinuity**. • **At $x = 2$**: Both left and right limits = 2, and $f(2) = 2$. Hence, the function is **continuous**. • **At $x = 3$**: Left and right limits = 1, and $f(3) = 1$. Therefore, the function is **continuous**. **Discontinuities:** $x = 0$ and $x = 1$ **Final Answer:** $\boxed{A}$	The solution examines the graph and claims vertical asymptotes at: • $x = 0$ • $x = 2$ It assumes these are discontinuities where the function is undefined and compares to options: • (A) $x = 0, x = 1$ • (D) $x = 2$ The reasoning concludes that both (A) and (D) seem correct and chooses: **Final Answer:** $\boxed{A}$

Table 3: Qualitative comparison of reasoning traces from our Qwen-2.5-VL-7B (SFT + DPO) model and the LLaVA-CoT baseline on two multimodal math problems. Each example includes the input image and question, followed by the model's reasoning trace and final answer. In the first example, our model correctly identifies the geometric structure and applies the Law of Cosines to compute the answer, while LLaVA-CoT incorrectly assumes a parallelogram and reaches an incorrect conclusion. In the second example, our model accurately identifies jump discontinuities based on left and right limits, whereas LLaVA-CoT misinterprets vertical asymptotes and includes a spurious discontinuity. These examples illustrate how difficulty-aware, compressed reasoning enables more accurate and structured outputs compared to baseline multimodal CoT models.