DreamPRM: Domain-Reweighted Process Reward Model for Multimodal Reasoning

Qi Cao
University of California, San Diego
q9cao@ucsd.edu

Ruiyi Wang
University of California, San Diego
ruiyi@ucsd.edu

Ruiyi Zhang
University of California, San Diego
ruz048@ucsd.edu

Sai Ashish Somayajula
University of California, San Diego
ssomayaj@ucsd.edu

Pengtao Xie
University of California, San Diego
Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)
p1xie@ucsd.edu

Abstract

Extending process reward models (PRMs) to multimodal LLMs is hindered by broad domain coverage, train–test distribution shift, and severe dataset quality imbalance. We propose DreamPRM, a bi-level, domain-reweighted framework: lower-level fine-tuning learns with per-domain weights to prioritize high-quality reasoning signals, while upper-level evaluation on a meta set updates these weights via an aggregation loss. Across diverse math reasoning benchmarks, DreamPRM consistently enhances state-of-the-art MLLMs and outperforms strong baselines in data selection and test-time scaling.

1 Introduction

Reasoning [47] has advanced LLMs [1, 7, 50, 41] in math problem solving, with Process Reward Models (PRMs) [25, 23] offering step-level supervision and guiding models toward the most valid reasoning trajectories. Given these successes, extending PRMs to multimodal LLMs (MLLMs) [60, 24] for visual math problem-solving is a natural next step. However, multimodal inputs combine high-dimensional visual signals with discrete language tokens, broadening the input space and worsening *distribution shifts* [48]. Moreover, multimodal reasoning datasets suffer from severe *quality imbalance* [66, 29], where noisy or trivial samples dilute effective training (Figure 1). As a result, naively transferring text-based PRM strategies [58, 33] underperforms due to poor generalization [9].

To address these challenges, we propose DreamPRM, a domain-reweighted bi-level optimization framework inspired by domain reweighting methods [45, 10, 49]. At the lower level, PRMs are fine-tuned across multiple datasets with learnable domain weights that emphasize high-quality domains and suppress noisy ones. At the upper level, meta-evaluation on a held-out dataset updates the weights via an aggregation loss [12, 27], improving robustness and generalization. Experiments on diverse multimodal reasoning benchmarks, including mathematical and general domains, demonstrate that DreamPRM consistently improves state-of-the-art MLLMs and outperforms alternative data-selection and test-time scaling strategies.

Our contributions are summarized as follows:

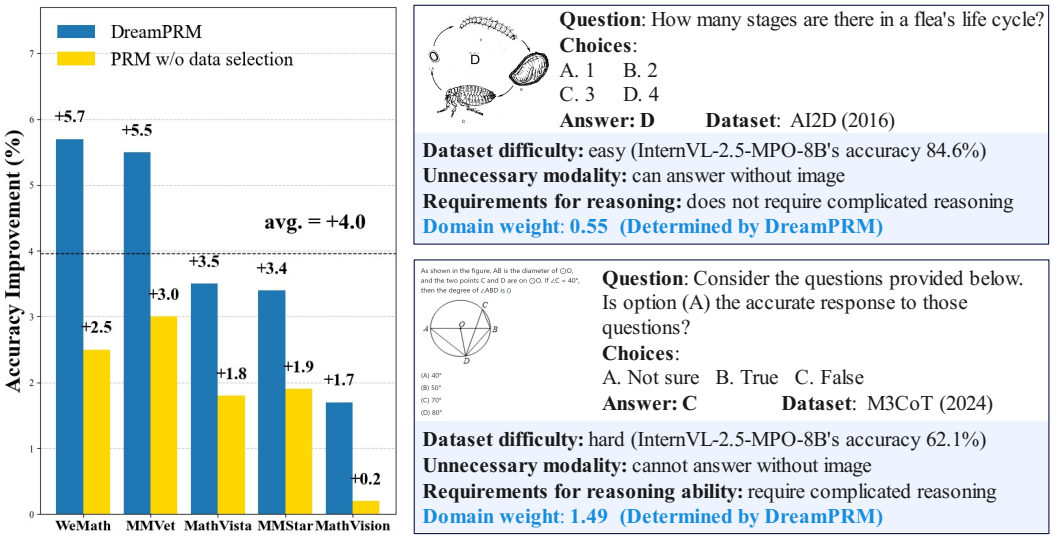


Figure 1: **DreamPRM improves multimodal reasoning by mitigating the dataset quality imbalance problem. Left**: On five benchmarks, DreamPRM outperforms base model (InternVL-2.5-8B-MPO [56]) by an average of $+4.0\%$. DreamPRM also consistently surpasses Vanilla PRM trained without data selection. **Right**: Easy AI2D [20] questions (weight 0.55) vs. hard M3CoT [5] questions (weight 1.49) shows how DreamPRM prioritizes data that demand deeper reasoning - samples requiring knowledge from both textual and visual modalities for step-by-step logical deduction.

- We propose DreamPRM, a *domain-reweighted* multimodal process reward model training framework that dynamically adjusts the importance of different training domains. We formulate the training process of DreamPRM as a *bi-level optimization* (BLO) problem, where the lower level optimizes the PRM via domain-reweighted fine-tuning, and the upper level optimizes domain weights with an aggregation function loss.
- We conduct extensive experiments using DreamPRM on a wide range of multimodal math benchmarks. Results indicate that DreamPRM consistently surpasses PRM baselines with other data selection strategies, confirming the effectiveness of its bi-level optimization based domain-reweighting strategy. Carefully designed evaluations further demonstrate that DreamPRM possesses both scaling capability and generalization ability to stronger models.

2 The Proposed Domain-reweighting Method

Overview. Training multimodal PRMs is difficult due to (1) dataset quality imbalance and (2) mismatch between training and inference (See Fig. 2). We propose **DreamPRM**, which learns domain importance via a novel aggregation loss that better simulates PRM inference. Under a bi-level framework, the lower level updates PRM parameters with domain-reweighted training, while the upper level optimizes domain weights on a meta dataset. An overview is shown in Fig. 3.

Notations. Let $\mathcal{I}$, $\mathcal{T}$, and $\mathcal{Y}$ denote the multimodal input space (images), textual instruction space, and response space, respectively. A multimodal large language model (MLLM) is formalized as a parametric mapping $M_\theta : \mathcal{T} \times \mathcal{I} \to \Delta(\mathcal{Y})$, where $\hat{y} \sim M_\theta(\cdot|x)$ represents the stochastic generation of responses conditioned on input pair $x = (t, I)$ including visual input $I \in \mathcal{I}$ and textual instruction $t \in \mathcal{T}$, with $\Delta(\mathcal{Y})$ denoting the probability simplex over the response space. We use $y \in \mathcal{Y}$ to denote the ground truth label from a dataset.

The process reward model (PRM) constitutes a sequence classification function $\mathcal{V}_\phi : \mathcal{T} \times \mathcal{I} \times \mathcal{Y} \to [0, 1]$, parameterized by ϕ, which quantifies the epistemic value of partial reasoning state $\hat{y}_i$ through scalar reward $p_i = \mathcal{V}_\phi(x, \hat{y}_i)$, modeling incremental utility toward solving instruction t under visual grounding I. Specifically, $\hat{y}_i$ represents the first i steps of a complete reasoning trajectory $\hat{y}$.

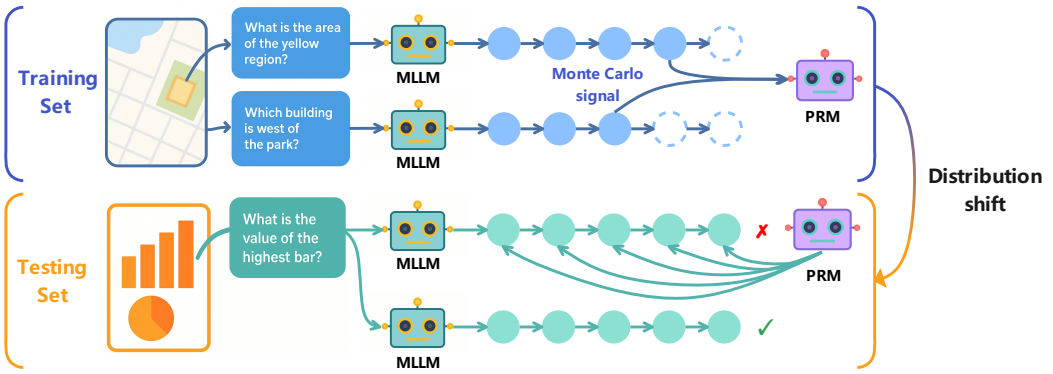


Figure 2: **General flow of training PRM and using PRM for inference. Training phase**: Train PRM with Monte Carlo signals from intermediate steps of Chain-of-Thoughts (CoTs). **Inference phase**: Use the trained PRM to verify CoTs step by step and select the best CoT. Conventional training of PRM has poor generalization capability due to *distribution shift* between training set and testing set.

Datasets. We use $K+1$ datasets, with K domains for training $\mathcal{D}_{\text{tr}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$ and one high-quality meta dataset $\mathcal{D}_{\text{meta}}$ for validation.

Lower-level optimization: domain-reweighted training of PRM. In lower-level optimization, we aim to update the weights ϕ of PRM with domain-reweighted training. We first define the typical PRM training loss $\mathcal{L}_{tr}$ on a single domain $\mathcal{D}_k$, given PRM parameters ϕ, as follows:

$$\mathcal{L}_{tr}(\mathcal{D}_k, \phi) = \sum_{(x,y)\in\mathcal{D}_k} \sum_{i=1}^{n} \mathcal{L}_{MSE}(\mathcal{V}_\phi(x, \hat{y}_i), p_i) \tag{1}$$

where $\hat{y}_i$ is the prefix of MLLM generated text $\hat{y} = M_\theta(x)$ given input pair $x = (t, I)$, and p_i is the process supervision signal value obtained by Monte Carlo estimation given input pair x, prefix $\hat{y}_i$ and ground truth label y, detailed in Appendix Equation 5. The PRM is optimized by minimizing the mean squared error (MSE) between supervision signal and PRM predicted score $\mathcal{V}_\phi(x, \hat{y}_i)$. With the PRM training loss on a single domain $\mathcal{D}_k$ above, we next define the domain-reweighted training objective of PRM on multiple training domains $\mathcal{D} = \{\mathcal{D}_k\}_{k=1}^K$. The overall objective is a weighted sum of the single-domain PRM training losses, allowing the contribution of each domain to be adjusted during the learning process:

$$\mathcal{L}_{tr}(\mathcal{D}_{tr}, \phi, \alpha) = \sum_{k=1}^{K} \alpha_k \mathcal{L}_{tr}(\mathcal{D}_k, \phi) \tag{2}$$

Here, $\alpha = \{\alpha_k\}_{k=1}^K$ represents the trainable domain weight parameters, indicating the importance of each domain. By optimizing this objective, we obtain the optimal value of PRM parameters ϕ^*:

$$\phi^*(\alpha) = \arg\min_{\phi} \mathcal{L}_{tr}(\mathcal{D}_{tr}, \phi, \alpha) \tag{3}$$

It is worth mentioning that only ϕ is optimized at this level, while α remains fixed.

Upper-level optimization: learning domain reweighting parameters. In upper-level optimization, we optimize the domain reweighting parameter α on meta dataset $\mathcal{D}_{meta}$ given optimal PRM weights $\phi^*(\alpha)$ obtained from the lower level. To make the meta learning target more closely reflect the actual PRM-based inference process, we propose a novel meta loss function $\mathcal{L}_{meta}$, different

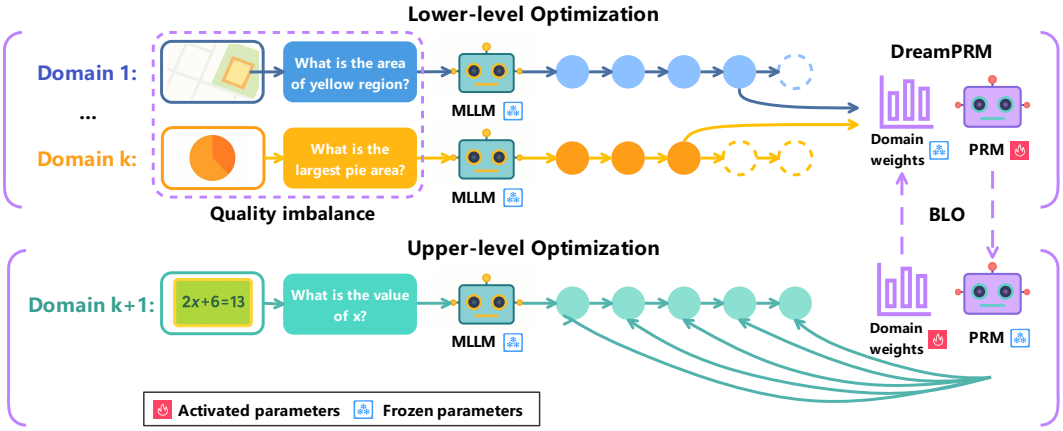


Figure 3: **DreamPRM framework. Lower-level:** PRM parameters are updated with domain weights. **Upper-level:** Domain weights are optimized on a meta dataset via an aggregation loss. This addresses dataset imbalance and improves generalization.

from the training loss $\mathcal{L}_{tr}$. Specifically, we first obtain an aggregated score $\mathcal{A}(p)$ for each generated solution $\hat{y}$ from the MLLM given input pair $x = (t, I)$. We then create a ground truth signal $r(\hat{y}, y)$ by assigning it a value of 1 if the generated $\hat{y}$ contains ground truth y, and 0 otherwise. The meta loss is defined as the mean squared error between aggregated score and ground truth signal:

$$\mathcal{L}_{meta}(\mathcal{D}_{meta}, \phi^*(\alpha)) = \sum_{(x,y)\in\mathcal{D}_{meta}} \mathcal{L}_{MSE}(\sigma(\mathcal{A}(\mathcal{V}_{\phi^*(\alpha)}(x, \hat{y}))), r(\hat{y}, y)) \tag{4}$$

where $\mathcal{A}$ represents the aggregation function (detailed in Appendix Equation 6), and σ denotes the sigmoid function to map the aggregated score to a probability.

3 Main Results

In this section, we apply DreamPRM to enhance math problem solving in MLLMs. We evaluate its performance with Best-of-N selection, sampling 8 reasoning chains per question, letting PRM select the best, and reporting gains over pass@1. Detailed experiment setting can be found at Appendix D

3.1 Benchmark evaluation

Tab. 1 summarizes the main results. **(1) DreamPRM consistently outperforms other PRM-based methods**, achieving 2%–3% gains over vanilla PRM and surpassing heuristic-based methods (s1-PRM, CaR-PRM), showing that handcrafted data selection is suboptimal while our automatic domain reweighting is more effective. **(2) DreamPRM also outperforms much larger closed-source MLLMs**, surpassing GPT-4v and Gemini-1.5-Pro on 4 of 5 datasets, and improving InternVL-2.5-8B-MPO by +4% on average. **(3) DreamPRM outperforms other test-time scaling methods**, since a high-quality PRM provides finer step-level supervision, leading to stronger gains.

3.2 Scaling and generalization analysis of DreamPRM

Scaling with more CoT candidates. Fig. 4 (left) shows that DreamPRM accuracy steadily improves on all five benchmarks as k increases from 2 to 8, expanding the radar plot outward. This indicates DreamPRM can robustly rank CoTs even under larger, more complex candidate pools.

Transfer to stronger base MLLMs. Fig. 4 (right) reports MATHVISTA accuracy with GPT-4.1-mini [39] and o4-mini [38]. For o4-mini, pass@1 improves from 80.6% to 85.2% at k=8, surpassing prior SOTA. Similar best-of-N trends on both models confirm DreamPRM's strong generalization.

Table 1: **Comparative evaluation of DreamPRM and baselines on multimodal math benchmarks. Bold numbers** indicate the best performance, while underlined numbers indicate the second best. The table reports accuracy (%) on WeMath, MathVista, MathVision, MMVet, and MMStar.

	WeMath *(loose)*	MathVista *(testmini)*	MathVision *(test)*	MMVet *(v1)*	MMStar *(test)*
Zero-shot Methods					
Gemini-1.5-Pro [42]	46.0	63.9	19.2	64.0	59.1
GPT-4v [39]	51.4	49.9	21.7	**67.7**	62.0
LLaVA-OneVision-7B [22]	44.8	63.2	18.4	57.5	61.7
Qwen2-VL-7B [55]	42.9	58.2	16.3	62.0	60.7
InternVL-2.5-8B-MPO [56]	51.7	65.4	20.4	55.9	58.9
Test-time Scaling Methods (InternVL-2.5-8B-MPO based)					
Self-consistency [57]	56.4	67.1	20.7	57.4	59.6
Self-correction [15]	54.0	63.8	21.6	54.9	59.7
ORM [44]	56.9	65.3	20.5	55.9	60.1
Vanilla PRM [25]	54.2	67.2	20.6	58.9	60.8
CaR-PRM [14]	54.7	67.5	21.0	60.6	61.1
s1-PRM [37]	57.1	65.8	20.2	60.1	60.4
DreamPRM (ours)	**57.4**	**68.9**	**22.1**	61.4	**62.3**

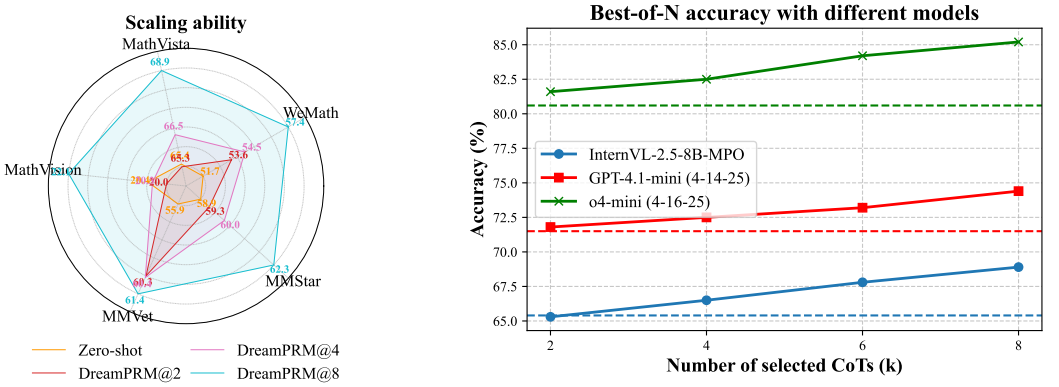


Figure 4: **(a)** Radar chart on five benchmarks: DreamPRM shows monotonic accuracy gains as selected CoTs increase (@2, @4, @8) over pass@1. **(b)** Best-of-N curves on MathVista for InternVL-2.5-8B-MPO (blue), GPT-4.1-mini (red), and o4-mini (green) show DreamPRM-ranked CoTs generalize across models, surpassing pass@1 (dashed) as k grows.

4 Conclusions

We introduce DreamPRM, the first domain-reweighted PRM for multimodal math problem solving. Using bi-level optimization to learn domain weights, DreamPRM alleviates dataset quality imbalance and improves generalization. Experiments on five benchmarks show consistent gains over vanilla and heuristic PRMs.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[2] Shuaichen Chang, David Palzer, Jialin Li, Eric Fosler-Lussier, and Ningchuan Xiao. Mapqa: A dataset for question answering on choropleth maps, 2022.

[3] Jiaqi Chen, Tong Li, Jinghui Qin, Pan Lu, Liang Lin, Chongyu Chen, and Xiaodan Liang. Unigeo: Unifying geometry logical reasoning via reformulating mathematical expression, 2022.

[4] 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.

[5] Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M^3cot: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought, 2024.

[6] Sang Keun Choe, Willie Neiswanger, Pengtao Xie, and Eric Xing. Betty: An automatic differentiation library for multilevel optimization. In *The Eleventh International Conference on Learning Representations*, 2023.

[7] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[9] Guanting Dong, Chenghao Zhang, Mengjie Deng, Yutao Zhu, Zhicheng Dou, and Ji-Rong Wen. Progressive multimodal reasoning via active retrieval, 2024.

[10] Simin Fan, Matteo Pagliardini, and Martin Jaggi. Doge: Domain reweighting with generalization estimation, 2024.

[11] Simin Fan, Matteo Pagliardini, and Martin Jaggi. DOGE: Domain reweighting with generalization estimation. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 12895–12915. PMLR, 21–27 Jul 2024.

[12] Chelsea Finn, P. Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

[13] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, and Lingpeng Kong. G-llava: Solving geometric problem with multi-modal large language model, 2023.

[14] Yuan Ge, Yilun Liu, Chi Hu, Weibin Meng, Shimin Tao, Xiaofeng Zhao, Hongxia Ma, Li Zhang, Boxing Chen, Hao Yang, Bei Li, Tong Xiao, and Jingbo Zhu. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation, 2024.

[15] Jiayi He, Hehai Lin, Qingyun Wang, Yi Fung, and Heng Ji. Self-correction is more than refinement: A learning framework for visual and language reasoning tasks, 2024.

[16] Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanwei Li, Yu Qi, Xinyan Chen, Liuhui Wang, Jianhan Jin, Claire Guo, Shen Yan, Bo Zhang, Chaoyou Fu, Peng Gao, and Hongsheng Li. Mme-cot: Benchmarking chain-of-thought in large multimodal models for reasoning quality, robustness, and efficiency, 2025.

[17] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering, 2018.

[18] Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Akos Kadar, Adam Trischler, and Yoshua Bengio. Figureqa: An annotated figure dataset for visual reasoning, 2018.

[19] Mehran Kazemi, Hamidreza Alvari, Ankit Anand, Jialin Wu, Xi Chen, and Radu Soricut. Geomverse: A systematic evaluation of large models for geometric reasoning, 2023.

[20] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images, 2016.

[21] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213, 2022.

[22] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer, 2024.

[23] Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making large language models better reasoners with step-aware verifier, 2023.

[24] Zongxia Li, Xiyang Wu, Hongyang Du, Fuxiao Liu, Huy Nghiem, and Guangyao Shi. A survey of state of the art large vision language models: Alignment, benchmark, evaluations and challenges. *arXiv preprint arXiv:2501.02189*, 2025.

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

[26] Adam Dahlgren Lindström and Savitha Sam Abraham. Clevr-math: A dataset for compositional language, visual and mathematical reasoning, 2022.

[27] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*, 2019.

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[29] 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. In *International Conference on Learning Representations (ICLR)*, 2024.

[30] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Intergps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *The 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

[31] 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, 2022.

[32] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning, 2022.

[33] Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. Improve mathematical reasoning in language models by automated process supervision, 2024.

[34] Qianli Ma, Haotian Zhou, Tingkai Liu, Jianbo Yuan, Pengfei Liu, Yang You, and Hongxia Yang. Let's reward step by step: Step-level reward model as the navigators for reasoning, 2023.

[35] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning, 2022.

[36] Minesh Mathew, Viraj Bagal, Rubèn Pérez Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V Jawahar. Infographicvqa, 2021.

[37] 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, 2025.

[38] OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024.

[39] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue,

Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

[40] Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Yifan Zhang, Xiao Zong, Yida Xu, Muxi Diao, Zhimin Bao, Chen Li, and Honggang Zhang. We-math: Does your large multimodal model achieve human-like mathematical reasoning?, 2024.

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

[42] Alex Reid et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens, 2024.

[43] Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. Solving geometry problems: Combining text and diagram interpretation. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1466–1476, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[44] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

[45] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting, 2019.

[46] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[47] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024.

[48] Shezheng Song, Xiaopeng Li, Shasha Li, Shan Zhao, Jie Yu, Jun Ma, Xiaoguang Mao, and Weimin Zhang. How to bridge the gap between modalities: Survey on multimodal large language model, 2025.

[49] Daouda Sow, Herbert Woisetschläger, Saikiran Bulusu, Shiqiang Wang, Hans-Arno Jacobsen, and Yingbin Liang. Dynamic loss-based sample reweighting for improved large language model pretraining, 2025.

[50] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[51] Chaojie Wang, Yanchen Deng, Zhiyi Lyu, Liang Zeng, Jujie He, Shuicheng Yan, and Bo An. Q*: Improving multi-step reasoning for llms with deliberative planning, 2024.

[52] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset, 2024.

[53] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[54] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[55] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[56] Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, and Jifeng Dai. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *arXiv preprint arXiv:2411.10442*, 2024.

[57] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023.

[58] Zihan Wang, Yunxuan Li, Yuexin Wu, Liangchen Luo, Le Hou, Hongkun Yu, and Jingbo Shang. Multi-step problem solving through a verifier: An empirical analysis on model-induced process supervision, 2024.

[59] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[60] Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S. Yu. Multimodal large language models: A survey, 2023.

[61] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[62] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step, 2024.

[63] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step, 2025.

[64] Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *The Thirteenth International Conference on Learning Representations*, 2025.

[65] Tianyu Yu, Haoye Zhang, Qiming Li, Qixin Xu, Yuan Yao, Da Chen, Xiaoman Lu, Ganqu Cui, Yunkai Dang, Taiwen He, Xiaocheng Feng, Jun Song, Bo Zheng, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Rlaif-v: Open-source ai feedback leads to super gpt-4v trustworthiness. *arXiv preprint arXiv:2405.17220*, 2024.

[66] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities, 2024.

[67] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024.

[68] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[69] Haojie Zheng, Tianyang Xu, Hanchi Sun, Shu Pu, Ruoxi Chen, and Lichao Sun. Thinking before looking: Improving multimodal llm reasoning via mitigating visual hallucination, 2024.

Appendix

A Related Works

Multimodal Reasoning Recent studies have demonstrated that incorporating Chain-of-Thought (CoT) reasoning [59, 21, 68] into LLMs encourages a step-by-step approach, thereby significantly enhancing question-answering performance. However, it has been reported that CoT prompting can't be easily extended to MLLMs, mainly due to hallucinated outputs during the reasoning process [56, 69, 16]. Therefore, some post-training methods have been proposed for enhancing reasoning capability of MLLMs. InternVL-MPO [56] proposes a mixed preference optimization that jointly optimizes preference ranking, response quality, and response generation loss to improve the reasoning abilities. Llava-CoT [62] creates a structured thinking fine-tuning dataset to make MLLM to perform systematic step-by-step reasoning. Some efforts have also been made for inference time scaling. RLAIF-V [65] proposes a novel self-feedback guidance for inference-time scaling and devises a simple length-normalization strategy tackling the bias towards shorter responses. AR-MCTS [9] combines Monte-Carlo Tree Search (MCTS) and Retrival Augmented Generation (RAG) to guide MLLM search step by step and explore the answer space.

Process Reward Model Process Reward Model (PRM) [25, 23, 34, 51] provides a more finer-grained verification than Outcome Reward Model (ORM) [8, 44], scoring each step of the reasoning trajectory. However, a central challenge in designing PRMs is obtaining process supervision signals, which require supervised labels for each reasoning step. Current approaches typically depend on costly, labor-intensive human annotation [25], highlighting the need for automated methods to improve scalability and efficiency. Math-Shepherd [53] proposes a method utilizing Monte-Carlo estimation to provide hard labels and soft labels for automatic process supervision. OmegaPRM [33] proposes a Monte Carlo Tree Search (MCTS) for finer-grained exploration for automatical labeling. MiPS [58] further explores the Monte Carlo estimation method and studies the aggregation of PRM signals.

Domain Reweighting Domain reweighting methodologies are developed to modulate the influence of individual data domains, thereby enabling models to achieve robust generalization. Recently, domain reweighting has emerged as a key component in large language model pre-training, where corpora are drawn from heterogeneous sources. DoReMi [61] trains a lightweight proxy model with group distributionally robust optimization to assign domain weights that maximize excess loss relative to a reference model. DOGE [11] proposes a first-order bi-level optimization framework, using gradient alignment between source and target domains to update mixture weights online during training. Complementary to these optimization-based approaches, Data Mixing Laws [64] derives scaling laws that could predict performance under different domain mixtures, enabling low-cost searches for near-optimal weights without proxy models. In this paper, we extend these ideas to process supervision and introduce a novel bi-level domain-reweighting framework.

B Problem Setting and Preliminaries

PRM training with Monte Carlo signals. Due to the lack of ground truth epistemic value for each partial reasoning state $\hat{y}_i$, training of PRM requires automatic generation of approximated supervision signals. An effective approach to obtain these signals is to use the Monte Carlo method [58, 54]. We first feed the input question-image pair $x = (t, I)$ and the prefix solution $\hat{y}_i$ into the MLLM, and let it complete the remaining steps until reaching the final answer. We randomly sample multiple completions, compare their final answers to the gold answer y, and thereby obtain multiple correctness labels. PRM is trained as a sequence classification task to predict these correctness labels. The ratio of correct completions at the i-th step estimates the "correctness level" up to step i, which is used as the approximated supervision signals p_i to train the PRM. Formally,

$$p_i = \texttt{MonteCarlo}(x, \hat{y}_i, y) = \frac{\texttt{num(correct completions from } \hat{y}_i)}{\texttt{num(total completions from } \hat{y}_i)} \tag{5}$$

PRM-based inference with aggregation function. After training a PRM, a typical way of conducting PRM-based MLLM inference is to use aggregation function [58]. Specifically, for each candidate solution $\hat{y}$ from the MLLM, PRM will generate a list of predicted probabilities $p = \{p_1, p_2, ..., p_n\}$ accordingly, one for each step $\hat{y}_i$ in the solution. The list of predicted probabilities are then aggregated using the following function:

$$\mathcal{A}(p) = \sum_{i=1}^{n} \log \frac{p_i}{1 - p_i}. \tag{6}$$

The aggregated value corresponds to the score of a specific prediction $\hat{y}$, and the final PRM-based solution is the one with the highest aggregated score.

Bi-level optimization. Bi-level optimization (BLO) has been widely used in meta-learning [12], neural architecture search [27], and data reweighting [46]. A BLO problem is usually formulated as:

$$\min_{\alpha} \mathcal{U}(\alpha, \phi^*(\alpha)) \tag{7}$$

$$s.t. \phi^*(\alpha) = \arg\min_{\phi} \mathcal{L}(\phi, \alpha) \tag{8}$$

where $\mathcal{U}$ is the upper-level optimization problem (OP) with parameter α, and $\mathcal{L}$ is the lower-level OP with parameter ϕ. The lower-level OP is nested within the upper-level one, and the two OPs are mutually dependent.

C Additional Method Details

Optimization algorithm Directly solving the bi-level optimization problem can be computational prohibitive due to its nested structure. Following previous work [6], we use approximated algorithm with a few unrolling steps. For example, under one-step unrolling, the updating of PRM's weights can be expressed as:

$$\phi^{(t+1)} = \phi^{(t)} - \beta_1 \nabla_{\phi} \mathcal{L}_{tr}(\mathcal{D}_{tr}, \phi, \alpha) \tag{9}$$

where β_1 is the learning rate in lower level optimization. After obtaining the updated PRM parameter $\phi^{(t+1)}$ from Equation 9, the domain-reweighting parameter α is then updated as follows:

$$\alpha^{(t+1)} = \alpha^{(t)} - \beta_2 \nabla_{\alpha} \mathcal{L}_{meta}(\mathcal{D}_{meta}, \phi^*(\alpha)) \tag{10}$$

where β_2 is the learning rate for upper level optimization. The two optimization steps in Equation 9 and Equation 10 are conducted iteratively until convergence to get optimal PRM weights ϕ^* and optimal domain reweighting parameter α^*.

D Additional Results and Ablations

D.1 Dataset Details

To enhance the robustness of DreamPRM, we collect a diverse set of datasets in lower-level optimization, spanning multiple domains to ensure a comprehensive coverage of multimodal reasoning tasks, as reported in Tab. 2. The selected 15 multimodal datasets covers 4 major categories including science, chart, geometry and commonsense, with a wide range of task types (QA, OCR, spatial understanding). Additionally, we observe that for some questions, given the current structural thinking prompts, MLLMs consistently produce either correct or incorrect answers. Continuing to sample

Table 2: Multimodal datasets involved in the fine-tuning of DreamPRM, organized by task category.

Task	Dataset
Science	AI2D [20], ScienceQA [31], M3CoT [5]
Chart	ChartQA [35], DVQA [17], MapQA [2], FigureQA [18]
Geometry	Geo170k [13], Geometry3K [30], UniGeo [3], GeomVerse [19], GeoS [43]
Commonsense	IconQA [32], InfographicsVQA [36], CLEVR-Math [26]

such questions is a waste of computational resources. Inspired by the dynamic sampling strategy in DAPO [66], we propose a similar dynamic sampling technique for Monte Carlo estimation that focuses on prompts with varied outcomes to improve efficiency. After processing and sampling, the training datasets in lower-level $\mathcal{D}_{tr}$ have around 15k examples (1k per each of the 15 domains), while the meta dataset in the upper-level $\mathcal{D}_{meta}$ has around 1k validation examples from the MMMU [67] dataset.

D.2 Benchmark details

At evaluation time, we use five multimodal reasoning benchmarks for testing the capability of DreamPRM. WEMATH [40], MATHVISTA [29], and MATHVISION [52] focus more on math-related reasoning tasks and logic and critical thinking, while MMVET [66] and MMSTAR [4] focus more on real-life tasks that require common knowledge and general reasoning abilities.

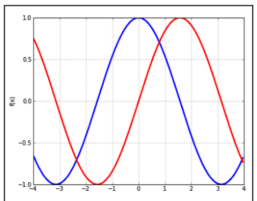
D.3 Experimental settings

Multistage reasoning. To elicit consistent steady reasoning responses from current MLLMs, we draw on the Llava-CoT approach [63], which fosters structured thinking prior to answer generation. Specifically, we prompt MLLMs to follow five reasoning steps: `(1) Restate the question. (2) Gather evidence from the image. (3) Identify any background knowledge needed. (4) Reason with the current evidence. (5) Summarize and conclude with all the information.` We also explore zero-shot prompting settings in conjunction with structural reasoning, which can be found in Appendix D.3. We use 8 different chain-of-thought reasoning trajectories for all test-time scaling methods, unless otherwise stated.

Structural Thinking Prompt The detailed structural thinking prompt applied in our experiments is reported in Fig. 5. We carefully design 5 reasoning steps to boost the reasoning capabilities of the MLLMs and enable process supervision.

5-step structural thinking for multimodal reasoning

You have been given a question that involves both an image and a text.
Your task is to analyze the question by following exactly five steps:
Step 1: Restate the question.
- Clearly rephrase or clarify the question in your own words.
Step 2: Gather evidence from the image.
- Describe any relevant visual details (e.g., objects, people, locations, interactions) that might address the question.
Step 3: Identify any background knowledge needed.
- Note any general facts, assumptions, or external knowledge that is necessary to address the question.
Step 4: Reason with the current evidence.
- Integrate the information from the image, text, and relevant background knowledge.
- Show how these pieces of evidence lead toward an answer.
Step 5: Summarize and conclude with all the information.
- Provide a concise, direct answer to the question, referencing the supporting evidence and reasoning.
Once you have completed your reasoning, provide your final answer in the format:
Final answer: ...



Question: Which function is monotonic in range [0, pi]?
Choices:
(A) the red one (B) the blue one
(C) both (D) none of them
Answer: (B) the blue one

Category: Math-targeted
Task: Textbook question answering
Context: Function plot
Grade: College
Math: Algebraic reasoning

MathVista

Figure 5: Zero-shot prompting for structural thinking.

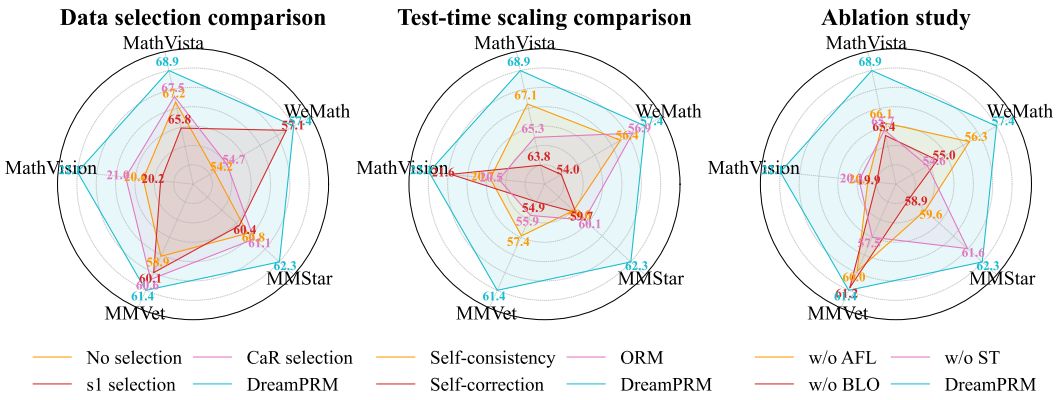


Figure 6: **Comparative evaluation of DreamPRM on multimodal reasoning benchmarks.** Radar charts report accuracy (%) on five datasets (WEMATH, MATHVISTA, MATHVISION, MMVET, and MMSTAR). **(a)** Impact of different data selection strategies. **(b)** Comparison with existing test-time scaling methods. **(c)** Ablation study of three key components, i.e. w/o aggregation function loss (AFL), w/o bi-level optimization (BLO), and w/o structural thinking (ST).

Base models. For inference, we use InternVL-2.5-8B-MPO [56] as the base MLLM, which has undergone post-training to enhance its reasoning abilities and is well-suited for our experiment. For fine-tuning PRM, we adopt Qwen2-VL-2B-Instruct [55]. Qwen2-VL is a state-of-the-art multimodal model pretrained for general vision-language understanding tasks. This pretrained model serves as the initialization for our fine-tuning process.

Training hyperparameters. In the lower-level optimization, we perform 5 inner gradient steps per outer update (unroll steps = 5) using the AdamW [28] optimizer with learning rate set to 5×10^{-7}. In the upper-level optimization, we use the AdamW optimizer ($\text{lr} = 0.01$, weight decay $= 10^{-3}$) and a StepLR scheduler (step size = 5000, $\gamma = 0.5$). In total, DreamPRM is fine-tuned for 10000 iterations. Our method is implemented with Betty [6], and the fine-tuning process takes approximately 10 hours on two NVIDIA A100 GPUs.

Baselines. We use three major categories of baselines: **(1)** State-of-the-art models on public leaderboards, including Gemini-1.5-Pro [42], GPT-4V [39], LLaVA-OneVision-7B [22], Qwen2-VL-7B [55]. We also carefully reproduce the results of InternVL-2.5-8B-MPO with structural thinking. **(2)** Test-time scaling methods (excluding PRM) based on the InternVL-2.5-8B-MPO model, including: (i) Self-consistency [57], which selects the most consistent reasoning chain via majority voting over multiple responses; (ii) Self-correction [15], which prompts the model to critically reflect on and revise its initial answers; and (iii) Outcome Reward Model (ORM) [44], which evaluates and scores the final response to select the most promising one. **(3)** PRM-based methods, including: (i) Vanilla PRM trained without any data selection, as commonly used in LLM settings [25]; (ii) s1-PRM, which selects high-quality reasoning responses based on three criteria - difficulty, quality, and diversity - following the s1 strategy [37]; and (iii) CaR-PRM, which filters high-quality visual questions using clustering and ranking techniques, as proposed in CaR [14].

D.4 Benchmark evaluation of DreamPRM

Tab. 1 presents the primary experimental results. We observe that: **(1) DreamPRM outperforms other PRM-based methods**, highlighting the effectiveness of our domain reweighting strategy. Compared to the vanilla PRM trained without any data selection, DreamPRM achieves a consistent performance gain of 2%-3% across all five datasets, suggesting that effective data selection is crucial for training high-quality multimodal PRMs. Moreover, DreamPRM also outperforms s1-PRM and CaR-PRM, which rely on manually designed heuristic rules for data selection. These results indicate that selecting suitable reasoning datasets for PRM training is a complex task, and handcrafted rules are often suboptimal. In contrast, our automatic domain-reweighting approach enables the model to adaptively optimize its learning process, illustrating how data-driven optimization offers a scalable solution to dataset selection challenges. **(2) DreamPRM outperforms SOTA MLLMs with**

much fewer parameters, highlighting the effectiveness of DreamPRM. For example, DreamPRM significantly surpasses two trillion-scale closed-source LLMs (GPT-4v and Gemini-1.5-Pro) on 4 out of 5 datasets. In addition, it consistently improves the performance of the base model, InternVL-2.5-8B-MPO, achieving an average gain of 4% on the five datasets. These results confirm that DreamPRM effectively yields a high-quality PRM, which is capable of enhancing multimodal reasoning across a wide range of benchmarks. **(3) DreamPRM outperforms other test-time scaling methods**, primarily because it enables the training of a high-quality PRM that conducts fine-grained, step-level evaluation. While most test-time scaling methods yield moderate improvements, DreamPRM leads to the most substantial gains, suggesting that the quality of the reward model is critical for effective test-time scaling.

D.5 Ablation study

In this section, we investigate the importance of three components in DreamPRM: **(1)** bi-level optimization, **(2)** aggregation function loss in upper-level, and **(3)** structural thinking prompt (detailed in Section D.3). As shown in the rightmost panel of Fig. 6, the complete DreamPRM achieves the best results compared to three ablation baselines across all five benchmarks. Eliminating bi-level optimization causes large performance drop (e.g., -3.5% on MATHVISTA and -3.4% on MMSTAR). Removing aggregation function loss leads to a consistent 1%-2% decline (e.g., 57.4% $\rightarrow$ 56.3% on WEMATH). Excluding structural thinking also degrades performance (e.g., -1.8% on MATHVISION). These results indicate that all three components are critical for DreamPRM to achieve the best performance. More detailed results are shown in Appendix Tab. 4.

Table 3: Accuracy on MATHVISTA using DreamPRM with varying numbers k of CoTs.

Model Name	pass@1	**DreamPRM (select k CoTs)**			
	k=1	k=2	k=4	k=6	k=8
InternVL-2.5-8B-MPO [56]	65.4	65.3	66.5	67.8	**68.9**
GPT-4.1-mini (4-14-25) [39]	71.5	71.8	72.5	73.2	**74.4**
o4-mini (4-16-25) [38]	80.6	81.6	82.5	84.2	**85.2**

Table 4: Ablation study evaluating the impact of individual components of DreamPRM

Ablation / Dataset	**WeMath**	**MathVista**	**MathVision**	**MMVet**	**MMStar**
DreamPRM (original)	**57.4**	**68.9**	**22.1**	**61.4**	**62.3**
w/o aggregation function loss	56.3 (-1.1)	66.1 (-2.8)	20.1 (-2.0)	60.0 (-1.4)	59.6 (-2.7)
w/o bi-level optimization	55.0 (-2.4)	65.4 (-3.5)	19.9 (-2.2)	61.2 (-0.2)	58.9 (-3.4)
w/o structural thinking	54.6 (-2.8)	65.7 (-3.2)	20.3 (-1.8)	57.5 (-3.9)	61.6 (-0.7)

D.6 Best-of-N results.

Tab. 3 reports the accuracy of two state-of-the-art models on MathVista dataset using DreamPRM with varying numbers k of CoTs. The results indicate that the performance scales well with the number of CoTs.

D.7 Analysis of learned domain weights

As shown in Fig. 7, final domain weights range from 0.55 to 1.49: M3COT [5] and FIGUREQA [18] have the highest values ($\sim$1.5), while AI2D [20] and ICONQA [32] are below 0.8. This pattern confirms that quality imbalance across datasets matters and is mitigated by DreamPRM.

D.8 Loss curves and domain weights.

The loss curves and domain weights during the fine-tuning of DreamPRM are illustrated in Fig. 8. It can be observed that the learnt distribution emphasizes informative mathematical figure domains while attenuating less relevant sources. Additionally, domain weights start at 1.0 and quickly diverge,

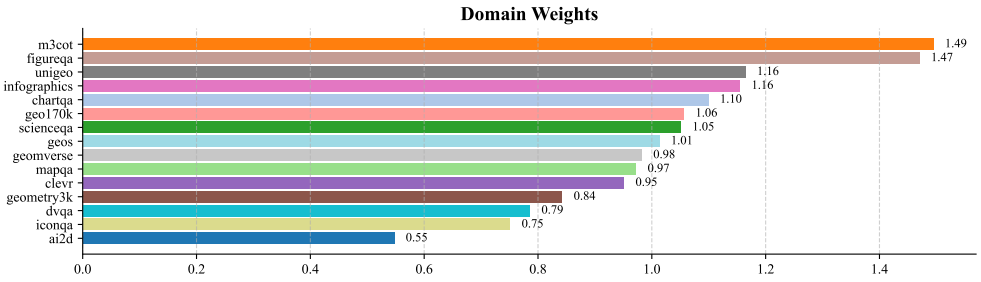


Figure 7: Learned domain weights after the convergence of the DreamPRM training process.

stabilizing after roughly half the training, and the inner and outer losses decrease steadily and plateau, indicating stable convergence of the bi-level training procedure.

D.9 Case study.

A complete case study illustrating DreamPRM's step-wise evaluation is reported in Fig. 9. DreamPRM assigns higher scores to high-quality, coherent reasoning steps, while penalizes flawed or unsupported steps.

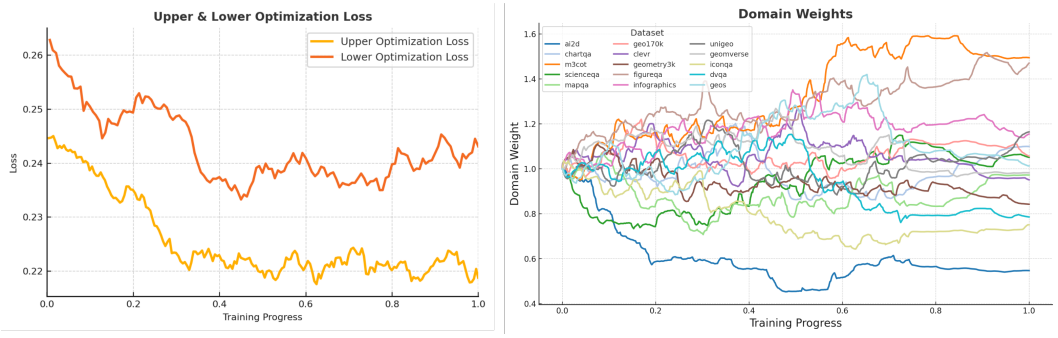
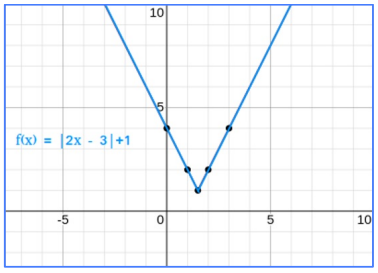


Figure 8: Optimization loss curves and dynamic domain weights throughout DreamPRM fine-tuning.

E Limitations & Future Work.

DreamPRM currently assumes a fixed set of domains and requires Monte-Carlo sampling, which can be computationally heavy. Future work could explore instance-level reweighting, adaptive sampling strategies, and integration with retrieval-augmented generation to further cut compute while broadening coverage. We will release code, trained weights, and evaluation scripts to facilitate reproducibility and community adoption.



Question:
The derivative of f(x) at x=2 is ____ that at x=5
Choices:
(A) larger than (B) equal to (C) smaller than
Ground Truth: B
Benchmark: MathVista

Step 1: Restate the question.
- The question is asking whether the derivative of the function \(f(x) = |2x - 3| + 1 \) at \(x = 2 \) is larger than, equal to, or smaller than the derivative at \(x = 5 \). **[DreamPRM: 0.628]**

Step 2: Gather evidence from the image.
- The image shows the graph of the function \(f(x) = |2x - 3| + 1 \).
- The graph has a V-shape, indicating it is an absolute value function. **[DreamPRM: 0.575]**

Step 3: Identify any background knowledge needed.
- The derivative of an absolute value function changes at the point where the expression inside the absolute value is zero.
- For \(f(x) = |2x - 3| + 1 \), the expression inside the absolute value is \(2x - 3 \).
- The derivative of \(|2x - 3| + 1 \) is \(2 \) for \(x > 1.5 \) and \(-2 \) for \(x < 1.5 \). **[DreamPRM: 0.598]**

Step 4: Reason with the current evidence.
- The point where the derivative changes is at \(x = 1.5 \).
- For \(x = 2 \), the function is in the interval where \(x > 1.5 \), so the derivative is \(2 \).
- For \(x = 5 \), the function is also in the interval where \(x > 1.5 \), so the derivative is \(2 \). **[DreamPRM: 0.748]**

Step 5: Summarize and conclude with all the information.
- Both the derivatives at \(x = 2 \) and \(x = 5 \) are \(2 \), so they are equal. **[DreamPRM: 0.812]**

Final answer: B ✓

Step 1: Restate the question.
- The question is asking how the derivative of \(f(x) \) at \(x = 2 \) compares to the derivative of \(f(x) \) at \(x = 5 \). **[DreamPRM: 0.676]**

Step 2: Gather evidence from the image.
- The image shows the graph of the function \(f(x) = |2x - 3| + 1 \).
- There are points marked on the graph at \(x = 2 \) and \(x = 5 \). **[DreamPRM: 0.499]**

Step 3: Identify any background knowledge needed.
- For absolute value functions, the derivative involves identifying where the function switches from decreasing to increasing.
- The function \(f(x) = |2x - 3| + 1 \) changes slope at the vertex of the absolute value, which is at \(x = \frac{3}{2} \). **[DreamPRM: 0.561]**

Step 4: Reason with the current evidence.
- At \(x = 2 \), the function appears linear with a negative slope (since it's to the left of the vertex).
- At \(x = 5 \), the function appears linear with a positive slope (since it's to the right of the vertex).
- The slope at \(x = 2 \) (negative) is steeper/more negative than the slope at \(x = 5 \) (positive). **[DreamPRM: 0.397]**

Step 5: Summarize and conclude with all the information.
- The derivative at \(x = 2 \) is negative and steeper than the derivative at \(x = 5 \), which is positive.
- Therefore, the derivative at \(x = 2 \) is larger in absolute value than the derivative at \(x = 5 \). **[DreamPRM: 0.396]**

Final answer: A ✗

Figure 9: A case study of DreamPRM's step-wise evaluation.