Diagram Formalization Enhanced Multi-Modal Geometry Problem Solver

Zeren Zhang[1,⋆], Jo-Ku Cheng[1,⋆], Jingyang Deng[1], Lu Tian[2], Jinwen Ma[1,✉],
Ziran Qin[3], Xiaokai Zhang[4], Na Zhu[4], Tuo Leng[4]

[1]School of Mathematical Sciences, Peking University, Beijing 100871, China
[2]01.AI, Beijing, China
[3]School of Electronic, Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
[4]School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
{eric_zhang, chengruogu, jingyang}@stu.pku.edu.cn, jwma@math.pku.edu.cn

Abstract—Mathematical reasoning remains an ongoing challenge for AI models, especially for geometry problems, which require both linguistic and visual signals. As the vision encoders of most MLLMs are trained on natural scenes, they often struggle to understand geometric diagrams, performing no better in geometry problem-solving than LLMs that only process text. This limitation is further amplified by the lack of effective methods for representing geometric relationships. To address these issues, we introduce the Diagram Formalization Enhanced Geometry Problem Solver (DFE-GPS), a new framework that integrates visual features, geometric formal language, and natural language representations. Specifically, we propose a novel synthetic data approach and construct a large-scale geometric dataset, SynthGeo228K, annotated with formal and natural language captions, designed to enhance the vision encoder to understand geometric structures better. Our framework improves MLLMs' ability to process geometric diagrams and extends their application to open-ended tasks on the formalgeo7k dataset.

Index Terms—multi-modal large language model, mathematical reasoning, geometry problem solver, geometric diagram formalization

I. INTRODUCTION

As large language models (LLMs) improve in mathematical reasoning [1]–[3], interest grows in how multi-modal large language models (MLLMs) can get better at processing visual information that aids mathematical comprehension [4], [5]. Solving geometry problems offers a practical way to evaluate MLLMs' mathematical reasoning abilities, as both visual diagrams and linguistic context are essential in this domain. This raises a fundamental question: Can MLLMs effectively understand geometric diagrams?

Studies indicate that MLLMs often struggle with geometric diagrams, showing no significant improvement over LLMs that only process textual information [6]. As shown in Fig. 1, we further find that the geometry problem-solving accuracy of MLLMs decreases when geometric diagrams are introduced, compared to scenarios where images are absent or blank. Some LLMs perform even better than MLLMs at solving geometry problems, because MLLMs have difficulty interpreting these diagrams, which results in MLLMs extracting incorrect or irrelevant information from them [7], [8].

Several reasons may explain this discrepancy. Most LLaVA-like architectures [9]–[11] use vision encoders that are pre-trained on natural scenes, which significantly differ from geometric diagrams [7]. Additionally, these models lack an effective methodology to represent geometric relationships.

To address these limitations, we introduce a new framework, the **D**iagram **F**ormalization **E**nhanced **G**eometry **P**roblem **S**olver (DFE-GPS), which incorporates a Diagram Formalizer — a model that leverages the formal language of these diagrams to improve the

⋆ Equal contribution.   ✉Corresponding author.
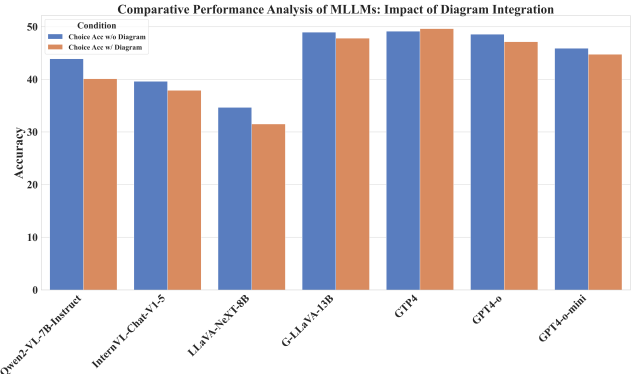Our model and dataset are available at https://github.com/zezeze97/DFE-GPS.

Fig. 1. Comparative performance analysis of MLLMs: Impact of diagram integration

visual components of the model and boost the LLM's recognition of geometric structures. We outline a pre-training task for the vision encoder focused on diagram formalization and introduce a new pipeline for generating 228K large-scale geometric diagrams with both geometric formal language, and natural language descriptions from scratch utilizing geometry model builder [12]. This approach addresses the limitation of insufficient and misaligned diagrams in pre-training [13]. By employing geometric formal language, the LLM can effectively master properties and relationships of geometric elements such as points, lines, and triangles and generate human-readable, step-by-step solutions. Consequently, our model can better comprehend geometric diagrams. We summarize the contributions as follows:

1) We introduce the DFE-GPS that integrates visual features, geometric formal language, and natural language, significantly enhancing geometric problem-solving as evidenced on the formalgeo7K test set.

2) We propose a novel synthetic data approach combining geometric diagrams with formalized language, optimizing vision encoder pre-training for better feature extraction from geometric diagrams. Additionally, we release the SynthGeo228k dataset.

3) We expand geometric problem-solving from multiple-choice questions to more challenging open-ended question answering, utilizing GPT-4o-mini to evaluate the step-wise problem-solving process with the Process Evaluation Score.

II. RELATED WORK

Geometry problem solving (GPS) has attracted substantial attention within the AI community. This field of study is divided into two GPS categories: single-modal, which relies solely on linguistic input, and multi-modal, which combines both vision and language.
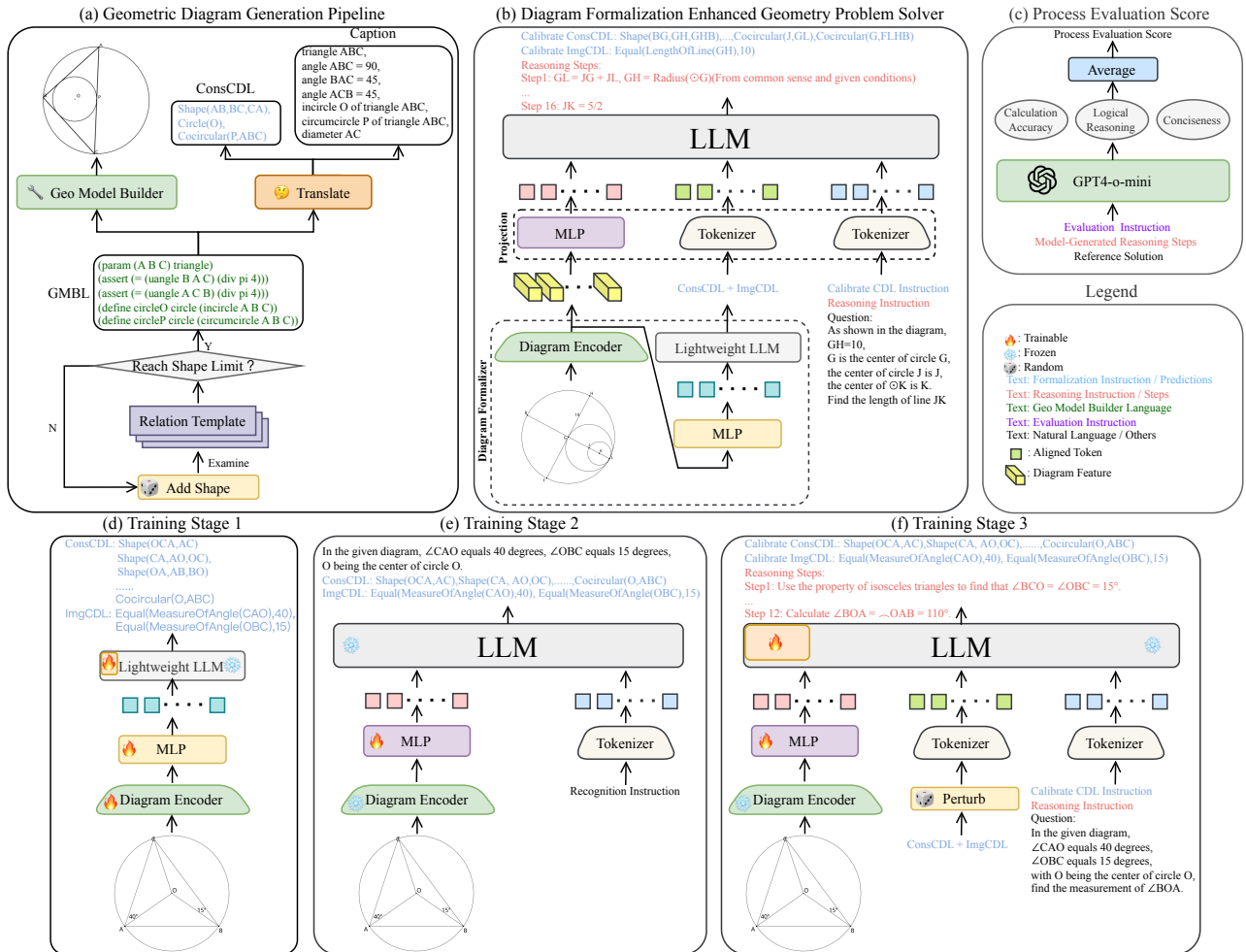
Fig. 2. Our proposed geometric diagram generation pipeline (a), diagram formalization enhanced geometry problem solver (b) and process evaluation score (c). The three-stage training overview is illustrated in (d, e, f).

Single-modal GPS Alpha Geometry [14] is a neuro-symbolic system that includes a symbolic deduction engine, achieving performance comparable to an average International Mathematical Olympiad (IMO) gold medalist. Similarly, FGeo-TP [15] utilizes the T5 language model to predict theorem sequences within its geometric formalized system FGPS, solving SAT-level geometric problems.

Multi-modal GPS requires a more complex data integration. This kind of research focuses primarily on generating theorem sequences to feed into a symbolic geometry solver. However, inconsistencies in dataset annotation standards have presented challenges in model training and performance evaluation [16]–[19].

InterGPS [20] converts text and diagrams into formal language for theorem prediction, but relies on a pre-constructed theorem knowledge base for final results. Also, PGPSNet [19] uses a Convolutional Neural Network (CNN) and a Gated recurrent units (GRU)-based model to generate solution programs, which require an executor for outputs.

Recent advancements have also explored the use of MLLMs for GPS. G-LLaVA [7] uses ChatGPT-3.5 to create an augmented dataset, Geo170K, and trains LLaVA without tuning the vision encoder. GeoGPT4V [13] uses GPT-4V to simplify complex problems into geometric QA pairs and generates corresponding images. However, it relies on the pre-existing questions from the dataset and cannot independently create diagrams.

III. PROPOSED APPROACH

This section begins with a review of the geometric formal language used in our approach, followed by an overview of the proposed geometric diagram generation pipeline. We then introduce the Diagram Formalization Enhanced Geometry Problem Solver (DFE-GPS) and detail the calculation of the Process Evaluation Score.

A. Preliminary: Geometric Formal Language

The theory of geometry formalization establishes a comprehensive framework for plane geometry. Formalgeo7k [21] is annotated with Conditional Declaration Language (CDL), which encompasses construction CDL, text CDL, image CDL, and goal CDL. We utilize construction CDL (ConsCDL) and image CDL (ImgCDL) to represent geometric diagrams. ConsCDL conveys geometric structure information, including basic shapes, collinearity, and cocircularity, while ImgCDL offers geometric and algebraic relations, such as segment length and angle relationships.

B. Geometric Diagram Generation

Numerous methods exist for generating geometric diagrams, such as GeoGebra [22] and MATLAB [23], but these typically require human interaction and cannot produce a wide range of meaningful diagrams autonomously. In our geometric diagram generation process, illustration in Fig 2(a), we create templates expressed in Geometry Model Building Language [12] by adding new shapes.

These templates, which range from single shapes to combinations of three with different geometric relationships, include basic elements like points and lines, as well as various triangles, polygons, and circles. When incorporating new elements, we analyze their geometric relationships, exploring scenarios such as circle lines intersections, angle congruences, and various configurations. Each template is translated into the formal language, ConsCDL, accompanied by detailed natural language captions specifying the shapes and their geometric elements. Our dataset, comprising 462 templates, provides a comprehensive exploration of basic geometry through the relationships between points, lines, and shapes. The Geometry Model Builder inherently allows for generating diverse diagrams from a single template by modifying point positions and orientations, with rotation being a common data augmentation method. Consequently, we have successfully generated over 228,000 geometric diagrams, constituting our SynthGeo228K dataset.

C. Structure and Training Process of DFE-GPS

As shown in Fig. 2(b), our proposed DFE-GPS model integrates multiple modalities and comprises three main components: a Diagram Formalizer, a Projection module, and a LLM. Specifically, the LLM processes three types of inputs: diagram features $\mathcal{F}_D$ extracted by the Diagram Encoder, formal diagram language representations (ConsCDL and ImgCDL) produced by the Diagram Formalizer, and natural language inputs that include problem statements and instructions. The Projection module aligns this information within the LLM's semantic space, enabling effective integration of these diverse inputs. As a result, the LLM first calibrates the formal representations predicted by the Diagram Formalizer to gain a deeper understanding of the diagram and then generates reasoning steps for problem-solving. For implementation, we employ the pre-trained SigLIP [24] as the Vision Encoder, Qwen2-0.5B-Instruct [25] as the Lightweight LLM, and Yi-1.5-Chat [10] (9B or 34B) as the primary LLM.

The training process consists of three stages, each centered on auto-regressive generation tasks. Let $\mathcal{T}_{in}$ denote the input text, $\mathcal{D}$ represent the input diagram, and $\mathcal{T}_{tar}$ be the target output. The training goal is defined by the following loss function:

$$\mathcal{L} = -\sum_{i=1}^{N} \log p[\mathcal{T}_{tar}^{i} | \mathcal{T}_{tar}^{(<i)}, \mathcal{T}_{in}, \mathcal{D}], \tag{1}$$

where $p(\cdot)$ represents the multi-modal generative model, $\mathcal{T}_{tar}^{i}$ indicates the i-th token of the target sequence, and N is the length of the target output. Fig. 2 (d,e,f) illustrates the three-stage training process.

Stage-1: The first stage focuses on training the Diagram Formalizer module, aiming to generate formalized language descriptions that correspond to geometric diagrams. During this phase, the parameters of the Vision Encoder and part of the Lightweight LLM (through LoRA [26]) are trainable to improve the model's ability to extract visual features.

Stage-2: The second stage centers on training the Projection modules to align vision features with the LLM's semantic space. This is achieved by generating natural language descriptions and formalized expressions for the geometric diagrams. During this stage, the parameters of the Diagram Encoder and the LLM are frozen, and only the MLP parameters linking visual features to the language model are trainable.

Stage-3: In the third stage, instruction fine-tuning enables the model to calibrate formalized diagram representations and solve problems. The input consists of geometric diagrams, formalized descriptions with random perturbations that simulate Diagram Formalizer errors, accompanied by problem text and calibration and reasoning instructions. The model learns to calibrate ConsCDL and ImgCDL, and then generate coherent natural language reasoning to solve the problem. During this stage, the parameters of the Diagram Encoder remain fixed, while the MLP and LLM parameters are trainable. Full parameter tuning is applied to the 9B model, whereas LoRA tuning is employed for the 34B model.

D. Process Evaluation Score

In addition to verifying the correctness of the final answers generated by the GPS, evaluating the reasoning process is even more critical. However, since our model generates solution processes in natural language, traditional formal methods used for symbol-based models [16]–[18] are not applicable. To address this challenge, we propose an evaluation method that leverages LLMs to assess the generated solution process. As illustrated in Fig.2(c), we input the model-generated reasoning steps and the reference solution into GPT-4o-mini, following the provision of tailored evaluation instructions. The model then reviews each reasoning step and evaluates the process based on three criteria: calculation accuracy, logical coherence, and conciseness. The average of these scores constitutes the Process Evaluation Score (PES).

IV. EXPERIMENTAL RESULTS

A. Datasets and Implementation Details

We develop two datasets from formalgeo7k, namely formalgeo-structure774k, focusing on geometric diagram formalization, and formalgeo-reasoning238k, enhanced via LLM-driven data augmentation. We also create the SynthGeo228k dataset using the Geometry Model Builder. In Stage 1, the Diagram Formalizer is trained for 4 epochs on both formalgeo-structure774k and SynthGeo228k (batch size 128, LoRA rank 16). Stage 2 involves 1 epoch of training on formalgeo-structure774k (batch size 256). In Stage 3, the 9B model undergoes full-parameter fine-tuning on formalgeo-reasoning238k (batch size 128), while the 34B model is fine-tuned using LoRA (rank 128, batch size 128). The entire process is accelerated using 8 NVIDIA A800 GPUs.

B. Performance of Diagram Formalizer

Table I presents the performance results of the Diagram Formalizer on the formalgeo7k test set. The sentence-level accuracy metric assesses the correctness of individual compositional statements within the predicted CDL, whereas the full-expression accuracy metric evaluates the accuracy of the entire predicted CDL. Our Diagram Formalizer exhibits strong overall performance, with a significant improvement attributed to the incorporation of synthetic data. In contrast, excluding synthetic data leads to a considerable decline in performance, especially in terms of full-expression accuracy.

TABLE I
THE PERFORMANCE OF DIAGRAM FORMALIZER AND ITS VARIANTS ON THE FORMALGEO7K TEST SET.

Model	ConsCDL		ImgCDL	
	Sentence	Full	Sentence	Full
Diagram Formalizer	**90.25**	**72.29**	**92.88**	**84.38**
w/o Synth-Data	88.85	68.29	90.80	80.57

C. Performance of Geometry Problem Solving

Table II presents a comparative analysis of the problem-solving performance between state-of-the-art LMMs, MLLMs, and our proposed DFE-GPS on the formalgeo7k test set. Our evaluation significantly extends beyond multiple-choice questions to include the more demanding open-ended questions. In the multiple-choice mode, the

model generates a solution and selects the correct answer from the provided options. In contrast, in the open-ended mode, the model independently generates a solution without any provided choices. We assess the models' performance by calculating the accuracy of the final answers across both modes. Furthermore, we utilize GPT-4o-mini to evaluate the correctness of the reasoning steps produced by the models, employing the PES.

Overall, the DFE-GPS-34B model exhibits outstanding performance across all metrics, achieving an accuracy of 82.38% in the multiple-choice mode, 75.33% in the open-ended mode, and a process evaluation score of 79.07. Although our smaller 9B model scores slightly lower, it still outperforms most other models. The increased difficulty of the open-ended mode leads to a decrease in accuracy across most models. Notably, DeepSeek-Prover-V1.5 [27], Yi-VL-6B [10], and QwenVL [28] fail to comply with instructions in the multiple-choice mode, often not selecting any of the provided options, resulting in accuracy rates lower than random guessing (25%). Within the same model series, larger parameter models demonstrate superior performance, underscoring the benefits of increased model size. Furthermore, models specifically trained on mathematical data (Qwen2-Math [2], DeepSeek-Math [1], G-LLaVA [7]) generally outperform general-purpose models. An exception is DeepSeek-Prover-V1.5, which, despite being trained under the LEAN [29] framework, performs poorly when confronted with natural language inputs.

TABLE II
PERFORMANCE COMPARISON OF FORMAL ENHANCED GPS AGAINST LMMS AND MLLMS ON THE FORMALGEO7K TEST SET.

| Model | Acc | | PES |
	Choice	Open-ended	
Llama-3.1-8B-Instruct [30]	26.38	16.57	48.56
Llama-3.1-70B-Instruct	40.86	20.86	56.48
deepseek-math-7b-instruct [1]	36.28	21.62	51.57
DeepSeek-Prover-V1.5-SFT [27]	19.24	34.10	66.14
DeepSeek-Prover-V1.5-RL	20.38	34.67	65.50
Yi-1.5-9B-Chat [10]	33.43	28.48	60.63
Yi-1.5-34B-Chat	38.29	25.62	61.18
Qwen2-7B-Instruct [25]	36.95	25.71	61.26
Qwen2-Math-7B-Instruct [2]	56.38	42.95	71.12
Qwen2-Math-72B-Instruct	57.71	49.81	76.89
Yi-VL-6B [10]	10.00	1.81	31.60
Yi-VL-34B	31.88	3.24	37.50
QwenVL [28]	9.05	1.81	28.91
Qwen2-VL-7B-Instruct [31]	40.10	22.19	61.84
Phi-3-vision-128k-instruct [32]	39.24	14.00	49.47
InternVL-Chat-V1-5 [33]	37.90	16.29	51.47
LLaVA-NeXT-8B [34]	31.52	10.19	48.83
LLaVA-NeXT-72B	34.76	26.38	49.45
G-LLaVA-7B [7]	45.33	15.71	52.20
G-LLaVA-13B	47.81	16.19	52.78
GPT-4-turbo [35]	49.62	38.00	69.63
GPT-4o	47.14	41.62	73.69
GPT-4o-mini	44.76	40.00	69.23
DFE-GPS-9B	77.05	68.67	76.00
DFE-GPS-34B	**82.38**	**75.33**	**79.07**

Beyond the GPT-4 series, MLLMs generally underperform compared to LLMs, likely due to their limited ability to effectively utilize diagrams. To explore this issue further, we conduct an experiment comparing the multiple-choice accuracy of MLLMs with and without the inclusion of images (Table III). While evaluating our model, we also control for the potential influence of formalized diagram language. The results indicate that for most MLLMs, incorporating diagrams often reduces accuracy in solving geometry problems, with QwenVL exhibiting a particularly notable decline. In contrast, our model demonstrates improved accuracy with the inclusion of diagrams, suggesting that our pre-trained Diagram Encoder effectively extracts features from diagrams and enhances overall performance.

TABLE III
EFFECTIVENESS OF MLLMS IN UTILIZING DIAGRAMS FOR SOLVING GEOMETRIC PROBLEMS.

| Model | Choice Acc | | Δ |
	w/o D	w/ D	
QwenVL [28]	19.90	9.05	-10.85
Qwen2-VL-7B-Instruct [31]	43.90	40.10	-3.80
Phi-3-vision-128k-instruct [32]	39.05	39.24	0.19
InternVL-Chat-V1-5 [33]	39.62	37.90	-1.72
LLaVA-NeXT-8B [34]	34.67	31.52	-3.15
G-LLaVA-13B [7]	48.95	47.81	-1.14
GPT-4-turbo [35]	49.14	49.62	0.48
GPT-4o	48.57	47.14	-1.43
GPT-4o-mini	45.90	44.76	-1.14
DFE-GPS-9B w/o CDL	**66.10**	**72.95**	**6.85**

D. Ablation Study

We examine the effects of omitting either Stage 1 or Stage 2 during training (Table IV) and observe significant performance drops, highlighting the essential role of our three-stage training process. Notably, Stage 1 pre-training with the Diagram Formalizer is more crucial than the process Stage 2.

TABLE IV
EVALUATION OF DFE-GPS-9B ACROSS THREE TRAINING STAGES

| Training Stages | | | Acc | |
Stage 1	Stage 2	Stage 3	Choice	Open-ended
✓	✓	✓	**77.05**	**68.67**
✓		✓	75.14	65.05
	✓	✓	67.33	56.57

To investigate the impact of incorporating geometric formal language, we conduct experiments using three solution modes: (1) generating the solution directly without any CDL input; (2) incorporating the Diagram Formalizer's predicted CDL into our model to generate the solution; (3) building on the second approach by calibrating the predicted CDL before generating the solution. Table V demonstrates a progressive improvement in accuracy across these approaches, suggesting that additional geometric formalization enhances the LLM's understanding of the diagrams, resulting in more effective problem-solving.

TABLE V
IMPACT OF DIFFERENT INPUT AND OUTPUT ON OUR 9B MODEL'S PERFORMANCE ON THE FORMALGEO7K TEST SET.

| Input | | Output | | Acc | |
D&Q	Pred-CDL	Cali-CDL	Solution	Choice	Open-ended
✓			✓	72.95	64.76
✓	✓		✓	76.19	67.81
✓	✓	✓	✓	**77.05**	**68.67**

V. CONCLUSION

We have established the Diagram Formalization Enhanced Geometry Problem Solver (DFE-GPS) as a new and effective multimodal large language model to leverage both geometric diagram and formalized information for solving geometric problems, along with a novel data generation pipeline for diagram formalization. The experimental results on the formalgeo7k demonstrate its significant improvements. Future work may explore reinforcement learning and tree search strategy to further enhance its performance.

REFERENCES

[1] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. Li, Y. Wu, and D. Guo, "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

[2] Qwen-2 Math, "Qwen-2 math: Enhancing math reasoning for language models," https://qwenlm.github.io/blog/qwen2-math/.

[3] H. Luo, Q. Sun, C. Xu, P. Zhao, J. Lou, C. Tao, X. Geng, Q. Lin, S. Chen, and D. Zhang, "Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct," *arXiv preprint arXiv:2308.09583*, 2023.

[4] P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, and J. Gao, "Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts," *arXiv preprint arXiv:2310.02255*, 2023.

[5] K. Wang, J. Pan, W. Shi, Z. Lu, M. Zhan, and H. Li, "Measuring multimodal mathematical reasoning with math-vision dataset," *arXiv preprint arXiv:2402.14804*, 2024.

[6] J. Zhang, Z. Li, M. Zhang, F. Yin, C. Liu, and Y. Moshfeghi, "Geoeval: benchmark for evaluating llms and multi-modal models on geometry problem-solving," *arXiv preprint arXiv:2402.10104*, 2024.

[7] J. Gao, R. Pi, J. Zhang, J. Ye, W. Zhong, Y. Wang, L. Hong, J. Han, H. Xu, Z. Li *et al.*, "G-llava: Solving geometric problem with multi-modal large language model," *arXiv preprint arXiv:2312.11370*, 2023.

[8] T. Guan, F. Liu, X. Wu, R. Xian, Z. Li, X. Liu, X. Wang, L. Chen, F. Huang, Y. Yacoob *et al.*, "Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 375–14 385.

[9] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, 2024.

[10] A. Young, B. Chen, C. Li, C. Huang, G. Zhang, G. Zhang, H. Li, J. Zhu, J. Chen, J. Chang *et al.*, "Yi: Open foundation models by 01. ai," *arXiv preprint arXiv:2403.04652*, 2024.

[11] H. Lu, W. Liu, B. Zhang, B. Wang, K. Dong, B. Liu, J. Sun, T. Ren, Z. Li, Y. Sun *et al.*, "Deepseek-vl: towards real-world vision-language understanding," *arXiv preprint arXiv:2403.05525*, 2024.

[12] R. Krueger, J. M. Han, and D. Selsam, "Automatically building diagrams for olympiad geometry problems." in *CADE*, 2021, pp. 577–588.

[13] S. Cai, K. Bao, H. Guo, J. Zhang, J. Song, and B. Zheng, "Geogpt4v: Towards geometric multi-modal large language models with geometric image generation," *arXiv preprint arXiv:2406.11503*, 2024.

[14] T. H. Trinh, Y. Wu, Q. V. Le, H. He, and T. Luong, "Solving olympiad geometry without human demonstrations," *Nature*, vol. 625, no. 7995, pp. 476–482, 2024.

[15] Y. He, J. Zou, X. Zhang, N. Zhu, and T. Leng, "Fgeo-tp: A language model-enhanced solver for euclidean geometry problems," *Symmetry*, vol. 16, no. 4, p. 421, 2024.

[16] M. Sachan and E. Xing, "Learning to solve geometry problems from natural language demonstrations in textbooks," in *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, 2017, pp. 251–261.

[17] J. Chen, J. Tang, J. Qin, X. Liang, L. Liu, E. P. Xing, and L. Lin, "Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning," *arXiv preprint arXiv:2105.14517*, 2021.

[18] J. Chen, T. Li, J. Qin, P. Lu, L. Lin, C. Chen, and X. Liang, "Unigeo: Unifying geometry logical reasoning via reformulating mathematical expression," *arXiv preprint arXiv:2212.02746*, 2022.

[19] M.-L. Zhang, F. Yin, and C.-L. Liu, "A multi-modal neural geometric solver with textual clauses parsed from diagram," *arXiv preprint arXiv:2302.11097*, 2023.

[20] P. Lu, R. Gong, S. Jiang, L. Qiu, S. Huang, X. Liang, and S.-C. Zhu, "Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning," *arXiv preprint arXiv:2105.04165*, 2021.

[21] X. Zhang, N. Zhu, Y. He, J. Zou, Q. Huang, X. Jin, Y. Guo, C. Mao, Z. Zhu, D. Yue *et al.*, "Formalgeo: The first step toward human-like imo-level geometric automated reasoning," *arXiv preprint arXiv:2310.18021*, 2023.

[22] Geogebra, "Geogebra: Dynamic mathematics for everyone," 2024. [Online]. Available: https://www.geogebra.org/

[23] MathWorks, "Matlab," http://matlab.p2hp.com/, 2023.

[24] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," 2023.

[25] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang *et al.*, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, 2024.

[26] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[27] H. Xin, Z. Z. Ren, J. Song, Z. Shao, W. Zhao, H. Wang, B. Liu, L. Zhang, X. Lu, Q. Du, W. Gao, Q. Zhu, D. Yang, Z. Gou, Z. F. Wu, F. Luo, and C. Ruan, "Deepseek-prover-v1.5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search," *arXiv preprint arXiv:2408.08152*, 2024.

[28] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A frontier large vision-language model with versatile abilities," *arXiv preprint arXiv:2308.12966*, 2023.

[29] L. d. Moura and S. Ullrich, "The lean 4 theorem prover and programming language," in *Automated Deduction–CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings 28*. Springer, 2021, pp. 625–635.

[30] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[31] Q. Team, "Qwen2-vl: To see the world more clearly," 2024. [Online]. Available: https://qwenlm.github.io/blog/qwen2-vl/

[32] M. Abdin, S. A. Jacobs, A. A. Awan, J. Aneja, A. Awadallah, H. Awadalla, N. Bach, A. Bahree, A. Bakhtiari, H. Behl *et al.*, "Phi-3 technical report: A highly capable language model locally on your phone," *arXiv preprint arXiv:2404.14219*, 2024.

[33] Z. Chen, W. Wang, H. Tian, S. Ye, Z. Gao, E. Cui, W. Tong, K. Hu, J. Luo, Z. Ma *et al.*, "How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites," *arXiv preprint arXiv:2404.16821*, 2024.

[34] B. Li, K. Zhang, H. Zhang, D. Guo, R. Zhang, F. Li, Y. Zhang, Z. Liu, and C. Li, "Llava-next: Stronger llms supercharge multimodal capabilities in the wild," 5 2024. [Online]. Available: https://llava-vl.github.io/blog/2024-05-10-llava-next-stronger-llms/

[35] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.