MATHS: MULTIMODAL TRANSFORMER-BASED HUMAN-READABLE SOLVER

Yicheng Pan[1], *Zhenrong Zhang*[1], *Jiefeng Ma*[1], *Pengfei Hu*[1],
Jun Du[1, †], *Qing Wang*[1, †], *Jianshu Zhang*[2], *Dan Liu*[2], *Si Wei*[2]

[1] NERC-SLIP, University of Science and Technology of China
[2] iFlytek Research
✉{jundu, qingwang}@ustc.edu.cn

ABSTRACT

Multimodal mathematical reasoning has gained increasing attention in recent times. However, previous effective methods have not tried to reason in the form of natural language. In this paper, we introduce a model named MATHS (**M**ultimod**A**l **T**ransformer-based **H**uman-readable **S**olver) for visual arithmetic and geometry problems in multimodal mathematical reasoning tasks. Drawing inspiration from Multimodal Large Language Models (MLLMs), our approach involves generating problem-solving processes expressed in natural language, in order to leverage the inherent reasoning capabilities embedded within language models. To address the challenge of precise calculations for language models, our work proposes a Math-Constrained Generation (MCG) method to impose hard constraints on generated outputs. Extensive experiments demonstrate our model excels in visual arithmetic task, and achieves results that are either better or comparable to existing methods in geometry problems. Code is available at https://github.com/ycpNotFound/MATHS.

***Index Terms*—** Multimodal, Mathematics Reasoning, Controllable Text Generation

1. INTRODUCTION

Recently, there has been a growing interest in multimodal mathematical in the field of AI research. This includes visual arithmetic problems [1], geometry problems involving calculations [2, 3, 4] and proofs [3]. Furthermore, Multimodal Large Language Models (MLLMs) have emerged as a new rising research hotspot, which utilize powerful Large Language Models (LLMs) to perform a wide range of multimodal tasks [5]. Drawing inspiration from this development, we aim to optimize the logical reasoning process into the form of natural language description based on the existing language models, with the goal of fully exploiting their latent reasoning capabilities.

As examples of multimodal math reasoning shown in Figure 1, the visual arithmetic task entails inferring hidden relationships between the numbers within each panel and predicting the missing number as the answer in the last panel [1]. This task involves the execution of arithmetic operations on numbers represented in visual objects. Concerning geometry problems, the specific tasks may involve calculations on length, area, angle or other variables, as well as proving numerical or geometric relationships in diagrams. In comparison to visual arithmetic problems, geometric problems require more advanced abstract thinking abilities and often depend on geometric theorems.

To forecast human-readable mathematical solutions, language models may encounter hallucination issues [6, 7], generating incorrect or inconsistent solutions that violate mathematical principles. In response to this challenge, we propose

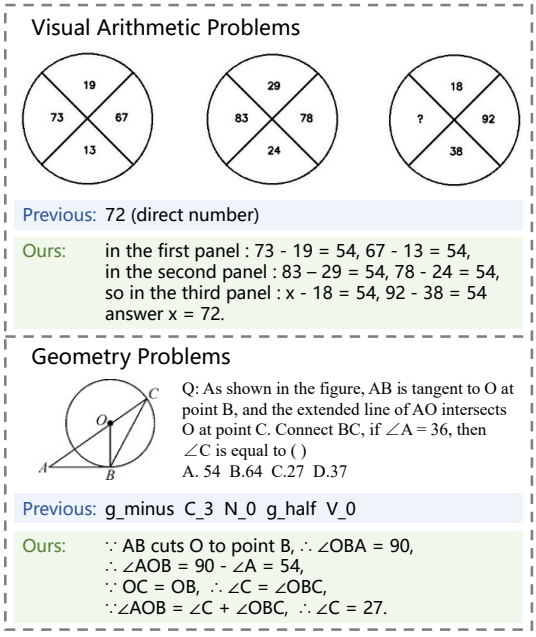

Fig. 1. Examples of multimodal mathematical reasoning, which illustrates samples in visual arithmetic problems and geometry problems. Rather than previous neural methods, our method provides human-readable solution processes.

† Corresponding author.

the Math-Constrained Generation (MCG) method. Through the external calculation module, our approach enables the imposition of stringent constraints during the generation process to uphold mathematical accuracy. Our method not only eliminates hallucination in calculations but also enables the model to utilize reasoning ability in higher-level problem-solving approaches.

Unifying multiple tasks in multimodal mathematical reasoning, we propose MATHS: **M**ultimod**A**l **T**ransformer-based **H**uman-readable **S**olver. MATHS provides a human-readable problem-solving process for diverse types of multimodal mathematical problems, and can be conveniently transferred to these tasks by using task-specific calculation module. We employ a pre-trained transformer encoder to extract visual information, which is subsequently passed as input to the cross-attention module of the transformer decoder layers. To enable the model to invoke the calculation module during generation, we introduce specialized tokens at specific locations in the solution sequences. We conduct extensive ablation experiments to demonstrate the efficacy of this approach.

Overall, our contributions can be summarized into three parts: (1) We unify diverse tasks in multimodal mathematics reasoning through the MATHS framework, a transformer-based model capable of generating problem-solving processes in natural language format. (2) We introduce a math-constrained generation method, ensuring the mathematical correctness of the generated content. (3) Experiments on existing datasets demonstrate that our method achieves outstanding performance in visual arithmetic problems and yields commendable results in geometry problems.

2. RELATED WORK

2.1. Multimodal Mathematics Reasoning

The field of multimodal mathematics reasoning encompasses various tasks, one of which is solving visual arithmetic problems. The evaluation of this task is done using the Machine Number Sense (MNS) dataset [1], which is created by And-Or-Graph (AOG) [8]. In the field of psychology, "number sense" provides an explanation of the cognitive process of numbers in humans. It is believed that the sense of numbers refers to the understanding of number concepts, proficiency in numerical operations, and the ability to solve mathematical problems flexibly [1]. Existing neural methods for visual arithmetic problems only predicted the final answer and overlooked the calculation process that involves rich mathematical information, leading to suboptimal performance.

Multimodal mathematics reasoning also includes geometry problems, such as calculation, proving [3] and diagram parsing [9]. Although neural methods have achieved results comparable to those of human in geometry, a symbolic solving sequence is required as the predicted label, which poses a challenge for human comprehension. Language models, particularly large language models (LLMs) [10, 11], commonly express solutions in natural language form. However, there have been no attempts to predict solutions in natural language for geometry problems. Furthermore, the task of diagram parsing presents difficulties for language models in performing detection and segmentation functions, so this work exclusively focuses on calculation and proving tasks.

2.2. Controllable Text Generation

Controllable Text Generation (CTG) [12] refers to the task involving generation of text by Pre-trained Language Models (PLMs) [13] according to the given controlled element. Based on traditional text generation, we can add control over attributes, styles, key information, etc. of the generated text, to ensure that it meets our expectations [14]. CTG methods for pre-trained language models typically require pre-training [15], fine-tuning [16], or training of an external discriminator [17]. Additionally, there also exist post-process methods that modify the appearance probability of words without further training [18]. Previous work on CTG often focuses on attribute-based generation, storytelling, or data-to-text tasks. However, the imposition of constraints to limit generation within mathematically correct ranges has not been fully studied yet. In this work, we utilize a math-constrained generation method inspired by post-process CTG methods, in order to align generated text with mathematical correctness.

3. METHODS

3.1. Overview

Our model solves multimodal math reasoning problems through an encoder-decoder framework, consisting of an image encoder and a language decoder with cross-attention, as shown in Figure 2. Such a structure enables the model to handle various tasks in multimodal math reasoning. Each task can be formulated as an image-to-text problem, where the input is an image $\mathbf{x}$ and the output is a text label $\mathbf{y}$ generated by the decoder with cross-attention. In cases where the inputs include text conditions, such as question text in geometry problems or task identification tokens, the label $\mathbf{y}$ can be divided into prompt $\mathbf{y}_p$ and target $\mathbf{y}_{target}$ components. Subsequently, the decoder receives $\mathbf{y}_p$ as prompts and predicts the answer $\mathbf{y}_{ans}$ in an autoregressive manner. Finally, we compute the loss function on $\mathbf{y}_{ans}$ only using the target $\mathbf{y}_{target}$.

3.2. Encoder-Decoder Framework

In this paragraph, we focus on our encoder-decoder framework. Instead of relying on external OCR engine [2] or textual clauses [4] to extract text information in diagrams, we utilize an end-to-end transformer encoder from Donut [19] that is pre-trained for OCR task. This encoder is used to extract both text symbols and visual features from an input image

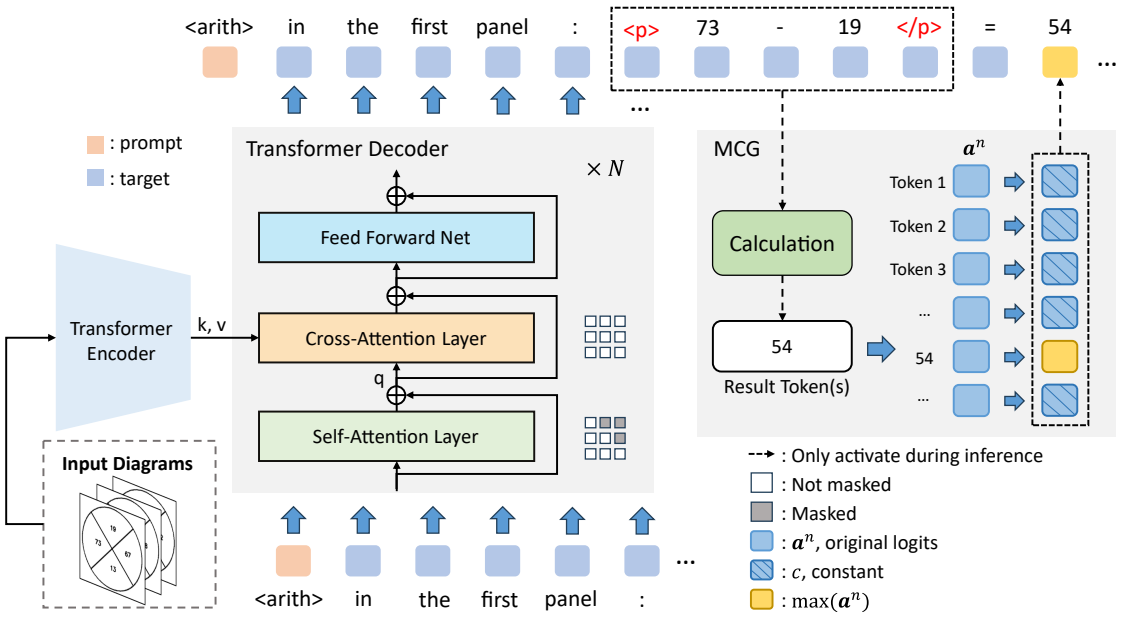

Fig. 2. Illustration of our MATHS framework using the MCG method. The MCG method is applied during inference. Specific tokens are marked in red font.

$\mathbf{x} \in \mathbb{R}^{H\times W\times C}$. The input image is encoded into a sequence of embeddings $\{\mathbf{z}_i|\mathbf{z}_i \in \mathbb{R}^d, 1 \le i \le n\}$, where n represents for number of image tokens and d stands for the hidden dimension of the encoder. On the other hand, the decoder is an auto-regressive transformer that interacts with visual features through cross-attention modules. These modules take visual features $\{\mathbf{z}\}$ as keys and values, and current decoder hidden states as queries. During training, we predict the next token on the condition of the visual features $\{\mathbf{z}\}$ and previous contexts. Specifically, we compute the probability of predicting answer tokens $\mathbf{y}_{ans}$ as follows:

$$p(\mathbf{y}_{ans}|\mathbf{z}, \mathbf{y}_p) = \prod_{i=1}^{L} p(y_i|\mathbf{z}, \mathbf{y}_{p,<i}, \mathbf{y}_{ans,<i}) \tag{1}$$

L is the length of total sequence of prompt and prediction $[\mathbf{y}_p, \mathbf{y}_{ans}]$, $\mathbf{y}_{p,<i}$ and $\mathbf{y}_{ans,<i}$ respectively represent for the prompt and answer tokens before the current prediction token y_i.

3.3. Math-Constrained Generation Method

To ensure the mathematical correctness of the generated processes, we propose the **M**ath-**C**onstrained **G**eneration (MCG) method. As illustrated in Figure 2, we introduce specific tokens to delineate the range of expressions in the training target. During inference, if the model generates the specific token at step n, a specific calculation program is invoked to compute the expression between the special tokens, producing result tokens denoted as $\mathbf{y}_{result}$. Subsequently, we maximize the the appearance probability of these result tokens in each following decoding step. Specifically, we set the output logits of the result tokens to be the maximum value among all logits, assigning a small constant value to the logits corresponding to other tokens. The process above can be formulated as:

$$\mathbf{a}_w^n = \begin{cases} \max(\mathbf{a}^n), & \textbf{if } w = y^n_{result} \\ c, & \textbf{otherwise} \end{cases} \tag{2}$$

$$p(y_n|\mathbf{z}, \mathbf{y}_{p,<n}, \mathbf{y}_{ans,<n}) = \text{softmax}(\mathbf{a}^n) \tag{3}$$

Among the above, $\mathbf{a}^n \in \mathbb{R}^V$ represents the output logits, $\mathbf{a}_w^n$ represents the logit of word w, and V is the vocabulary size of the language model. $p(y_n|\mathbf{z}, \mathbf{y}_{p,<n}, \mathbf{y}_{ans,<n})$ represents the probability distribution of the next token at step n. $y^n_{result} \in \mathbf{y}_{result}$ refers to the result token that should be set at step n. c is a constant set to be smaller than $\max(\mathbf{a}^n)$. Furthermore, the MCG method can be easily combined with a beam search strategy through maintaining a queue for each beam to record the calculation result tokens. Following the MCG method, we forces the correct calculation result to be the next prediction, while ensuring that other tokens do not to appear.

4. EXPERIMENTS

4.1. Datasets and Metrics

Previous studies on visual arithmetic problems conducted experiments on the synthetic dataset Machine Number Sense

System	Process	MCG	Context	Ans Acc	Proc Acc	Proc Acc / Ans Acc
T1				28.85	-	-
T2	✓			39.99	29.66	74.17
T3	✓	✓		40.13	34.31	85.51
T4	✓		✓	50.87	40.95	80.51
T5	✓	✓	✓	**64.98**	**57.30**	**88.18**

Table 1. Comparisons of different designed systems (with the same visual inputs) from T1 to T5 on MNS dataset. "Ans Acc" is the accuracy rate of final predicted answer. Positive samples of "Proc Acc" indecate that the processes, expressions and answer must all be correct. "Proc Acc / Ans Acc" represents the proportion of positive samples in "Proc Acc" among the positive samples in "Ans Acc".

(MNS) [1], which comprises 168k training samples, 56k validation samples, and 56k test samples. To compare with previous methods that assess accuracy through image classification, we define a successful solution as correctly predicting the answer. We also calculate the accuracy on various types of visual arithmetic problems, following previous literature. We use top-1 accuracy of final answer with beam search of size 10. Furthermore, it is considered detrimental if the model predicts the correct answer but provides an incorrect solution. Therefore, we evaluate the correctness of the generated intermediate process, which includes expressions and calculations. This evaluation involves assessing the models ability to list correct expressions, perform calculations correctly, and predict the right answer.

In the context of geometric problems, we assess several existing datasets. Among them, the UniGeo [3] dataset, which consists of 4,998 calculation problems and 9,543 proving problems, proves to be in alignment with our requirements, as it features annotations of solution processes in a natural language format. Additionally, we have also considered the PGPS9K [4] dataset, which provides high-quality diagrams and detailed annotations. However, its annotation of solutions is presented in the form of symbolic sequences, which poses challenges for translation into natural language. Therefore, We conduct experiments on the unified benchmark of both calculation and proving problems in the UniGeo dataset. We only compare the final calculated result with the target for the calculation questions. For proving questions, a predicted proof is considered correct only if every step is identical. We adopt top-10 accuracy with beam size of 10, following previous work [20, 3].

4.2. Implementation Details

Our model is implemented using Pytorch. We utilize a pre-trained Swin-Transformer [21] obtained from Donut [19] as the encoder. The decoder employs a BERT [13] architecture with cross-attention layers. We initialize component weights using pre-trained BERT weights, while the cross-attention module undergoes random initialization. Considering accelerated training, we use BERT-base architecture for ablation experiments and BERT-large for the final evaluation. We use the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The model is trained on 8 Nvidia Quadro RTX 6000 GPUs with a learning rate of $5e^{-5}$, following a linear learning rate decay with a warm-up strategy. The input image is resized to 300×300, and the our model is trained for 20 epochs on the MNS dataset with a total batch size of 64, and for 100 epochs on the UniGeo dataset with a total batch size of 32.

4.3. Ablation Study

Visual Context Learning. Since the model requires predicting the answer in the last panel of three panels, it is critical to utilize the contextual information of the first two panels. In order to validate this, we devise the systems T1, T2 and T4 as shown in Table 1. With the same visual inputs, T1 represents predicting direct answer without process, T2 only predict the expressions and answer of the last panel, and the prediction of T4 encompasses the entire process with the visual context from other panels. Results show that the accuracy rate of T1, T2, and T3 increases sequentially, which underlines the significant influence that information in visual context holds in reasoning.

Efficacy of MCG. We further explore the efficacy of our MCG method. Language models can sometimes generate incorrect solutions due to hallucinations in mathematical reasoning, while our MCG method can relieve this problem. These phenomena can be found in system T2, T3 and T4, T5 in Table 1. Utilizing the MCG method, T3 achieves a relatively slight improvement in accuracy compared to T2, while the accuracy of T5 increase significantly compared to T5. Furthermore, we design experiments to evaluate the correctness of intermediate solution processes. We also calculate the proportion of answer-correct samples in process-correct samples, which evaluates the credibility of the problem-solving process. Results show that methods employing MCG increase both accuracy and credibility of processes compared to methods without it, indicating the capacity of MCG to augment the dependability of the generated solutions.

Method	MNS							UniGeo	
	Combination		Composition		Partition		Mean	Calculation	Proving
	H	A	H	A	H	A			
GPT-4 [1]	35.00	14.66	3.33	11.95	21.79	21.59	16.40	-	-
GPT-4V [1]	48.00	29.33	5.33	20.40	26.92	31.25	24.50	5.0	74.0
Human [1]	66.82	93.64	61.36	78.18	77.27	88.18	77.58	-	-
Search [1]	**64.38**	56.08	29.81	61.84	**59.70**	67.59	56.70	-	-
ResNet [1]	27.90	24.22	23.42	23.73	26.61	27.78	25.29	-	-
NGS [20]	-	-	-	-	-	-	-	56.9	53.2
Geoformer [3]	-	-	-	-	-	-	-	**62.5**	56.4
MATHS (Ours)	63.51	**75.40**	**33.94**	**79.72**	52.81	**80.10**	**65.72**	57.0	**83.5**

[1] We test GPT-4 and GPT-4V on 1k samples in the MNS dataset and 100 samples in both calculation and proving task of UniGeo dataset.

Table 2. Performance of previous state-of-the-art methods and our method. Combination, composition and partition refer to different layout types of geometric elements. Holistic (H) and Analytic (A) represent distinct problem-solving styles. **Bold** indicates the SOTA and underline indicates the second best.

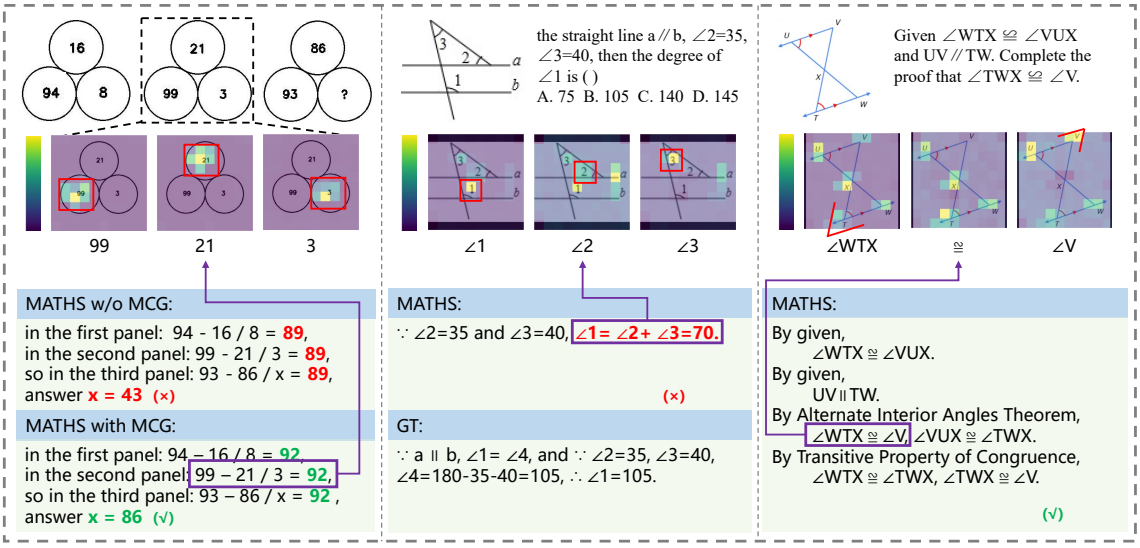

Fig. 3. Examples of outputs and cross attention maps. Special characters are omitted for ease of reading.

4.4. Comparison with State-of-the-art Methods

Table 2 presents the experimental results of our top-performing execution on the MNS dataset, where the performance of humans [1] and and results of previous neural-based and symbolic-based methods are also shown in. Search, which means context-guided search [1], takes numbers and semantic context information as input. The ResNet baseline predicts the answers from 0 to 99 according to input images. Our method outperforms both previous neural network and search-based methods on average. We also achieves excellent results in a majority of categories from the MNS dataset. However, it's relevant to note that humans perform better than our method on average and in some categories.

Table 2 also shows the results on the UniGeo dataset. It is observed that our method exhibited lower performance compared to existing state-of-the-art methods. It should be noted that our method focuses on generating contents of natural language form, and uses labels different from previous methods that utilize symbolic sequences. Therefore, the results are not entirely comparable. Besides, Geoformer [3] is pre-trained with carefully designed pre-training tasks, while our MATHS does not. However, experiments on this dataset can also demonstrate the transferability of our framework. We will leave the design of a universal pre-training strategy for multimodal mathematical reasoning for future work.

Our method have achieved advancements in solving proving problems. Besides the effectiveness of our model in natural language processing, this result can be attributed to the similarity of the word and diagram distributions in the training set and the test set, which indicates that the solutions in

the proving dataset may lack diversity for language models. Furthermore, the application of MCG to geometry problems is challenging, because the solutions of calculation problems almost do not involve calculation processes, and the proving questions do not require additional calculations either.

Additionally, we evaluate the performance of GPT-4 and GPT-4V providing images and instructions with one-shot prompting strategy. It can be observed that GPT-4V achieves a higher average accuracy than GPT-4 with text only on the MNS dataset. As for geometry problems, we only evaluate the performance of GPT-4V, which achieves poor results in calculation task and relatively comparable results in proving task. For specific analysis and prompt design, please refer to the supplementary materials.

4.5. Case Study

As illustrated in Figure 3, we present three instances to elucidate both the efficacy and shortcomings of the model. We also draw cross-attention maps at some important steps in the last layer of our model. In the first case of visual arithmetic problems, our MATHS without MCG predicts the correct expressions of three panels correctly, but outputs the wrong calculation results, indicated in bold red font. Through MCG, our model can correctly outputs the calculation results, and finally predicts the correct answer. In the second case of geometry calculation problem, our model make typical geometric mistakes violating geometric theorems, also shown in bold red font. Attention maps also illustrate that attention is paid to key positions by our model, but the generated reasoning process is wrong. In the third case of proving question, our model predict the correct proof, but the attention maps is not entirely in accordance with our expectations. It is still challenging for language models to align generated content with geometric diagrams and theorems.

5. CONCLUSION

In this work, we propose MATHS, a framework based on multimodal transformers that provides human-readable solutions for a range of multimodal mathematics reasoning tasks. We also present a math-constrained generation method that facilitates the production of precise and credible solutions. Experimental results demonstrate exceptional performance in visual arithmetic problems and commendable results in geometry problems, indicating the effectiveness of our approach. To further advance our work, we will strive to design effective multimodal pre-training strategies. Moreover, we aim to expand our paradigm of natural language problem-solving and math-constrained generation method to large language models (LLMs).

6. REFERENCES

[1] Wenhe Zhang, Chi Zhang, Yixin Zhu, and Song-Chun Zhu, "Machine number sense: A dataset of visual arithmetic problems for abstract and relational reasoning," in *AAAI*, 2020, pp. 1332–1340.

[2] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu, "Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning," in *ACL*, 2021.

[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," in *EMNLP*, 2022.

[4] Ming-Liang Zhang, Fei Yin, and Cheng-Lin Liu, "A multi-modal neural geometric solver with textual clauses parsed from diagram," in *IJCAI*, 2023.

[5] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen, "A survey on multimodal large language models," *arXiv preprint arXiv:2306.13549*, 2023.

[6] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al., "Siren's song in the ai ocean: A survey on hallucination in large language models," *arXiv preprint arXiv:2309.01219*, 2023.

[7] 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," *arXiv preprint arXiv:2310.02255*, 2023.

[8] Song-Chun Zhu, David Mumford, et al., "A stochastic grammar of images," *Foundations and Trends® in Computer Graphics and Vision*, vol. 2, no. 4, pp. 259–362, 2007.

[9] Ming-Liang Zhang, Fei Yin, Yi-Han Hao, and Cheng-Lin Liu, "Plane geometry diagram parsing," in *IJCAI*, 7 2022, pp. 1636–1643.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[11] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al., "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[12] Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov, "Exploring controllable text generation techniques," *arXiv preprint arXiv:2005.01822*, 2020.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song, "A survey of controllable text generation using transformer-based pre-trained language models," *ACM Computing Surveys*, vol. 56, no. 3, pp. 1–37, 2023.

[15] Nitish Shirish Keskar, Bryan McCann, Lav Varshney, Caiming Xiong, and Richard Socher, "CTRL - A Conditional Transformer Language Model for Controllable Generation," *arXiv preprint arXiv:1909.05858*, 2019.

[16] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan, "Dialogpt: Large-scale generative pre-training for conversational response generation," in *ACL, system demonstration*, 2020.

[17] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu, "Plug and play language models: A simple approach to controlled text generation," in *ICLR*, 2020.

[18] Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer, "A plug-and-play method for controlled text generation," in *Findings of EMNLP*, Punta Cana, Dominican Republic, 2021, pp. 3973–3997, Association for Computational Linguistics.

[19] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park, "Ocr-free document understanding transformer," in *ECCV*, 2022.

[20] Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric P Xing, and Liang Lin, "Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning," in *Findings of ACL*, 2021.

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021.