Math-PUMA: Progressive Upward Multimodal Alignment to Enhance Mathematical Reasoning

Wenwen Zhuang[1*], **Xin Huang**[2*], **Xiantao Zhang**[3*], **Jin Zeng**[1]
[1]University of Chinese Academy of Sciences
[2]Beijing Institute of Technology
[3]Beijing University of Aeronautics and Astronautics
zhuangwenwen23@mails.ucas.ac.cn, huangxin@bit.edu.cn, zhangxiantao@buaa.edu.cn, zengjing2she327@gmail.com

Abstract

Multimodal Large Language Models (MLLMs) excel in solving text-based mathematical problems, but they struggle with mathematical diagrams since they are primarily trained on natural scene images. For humans, visual aids generally enhance problem-solving, but MLLMs perform worse as information shifts from textual to visual modality. This decline is mainly due to their shortcomings in aligning images and text. To tackle aforementioned challenges, we propose Math-PUMA, a methodology focused on **P**rogressive **U**pward **M**ultimodal **A**lignment. This approach is designed to improve the mathematical reasoning skills of MLLMs through a three-stage training process, with the second stage being the critical alignment stage. We first enhance the language model's mathematical reasoning capabilities with extensive set of textual mathematical problems. We then construct a multimodal dataset with varying degrees of textual and visual information, creating data pairs by presenting each problem in at least two forms. By leveraging the Kullback-Leibler (KL) divergence of next-token prediction distributions to align visual and textual modalities, consistent problem-solving abilities are ensured. Finally, we utilize multimodal instruction tuning for MLLMs with high-quality multimodal data. Experimental results on multiple mathematical reasoning benchmarks demonstrate that the MLLMs trained with Math-PUMA surpass most open-source MLLMs. Our approach effectively narrows the performance gap for problems presented in different modalities.

Code — https://github.com/wwzhuang01/Math-PUMA

Introduction

Large Language Models (LLMs) have demonstrated remarkable reasoning capabilities, particularly when tackling mathematical problems in textual form (Wei et al. 2022; Chen et al. 2022; Gou et al. 2023; Yu et al. 2023; Shao et al. 2024). However, Multimodal Large Language Models (MLLMs) face greater challenges when tackling problems that involve images. These models need to not only interpret textual information but also comprehend mathematical diagrams and identify details crucial for solving problems. Although MLLMs have exhibited notable efficacy in general visual question answering (Radford et al. 2021; Li et al. 2022; Liu et al. 2023), their training predominantly relies on datasets comprising natural scene images. This reliance engenders a substantial domain discrepancy when these models are applied to mathematical diagrams, thereby resulting in inferior performance.

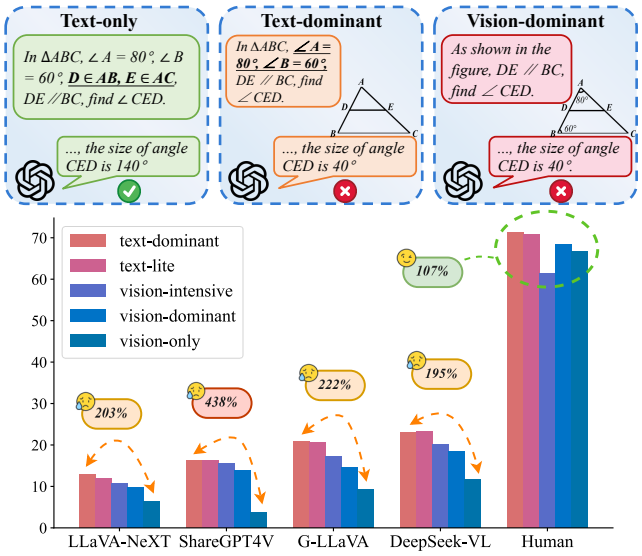

Figure 1: (*Top*) Three examples of GPT-4o solving multimodal math problems. These examples represent different modalities of the same question. (*Bottom*) Results of several open-source MLLMs and human on five different tasks of MATHVERSE (Zhang et al. 2024a).

For humans, regardless of the modality in which information is presented, problems with equivalent amounts of information tend to have similar levels of difficulty. Furthermore, incorporating images into problem-solving tasks can enhance human comprehension and resolution abilities. As illustrated in Figure 1, an increase in visual data often correlates with a decline in the efficacy of most MLLMs. Additionally, there is a notable disparity in effectiveness between text-centric and exclusively visual problems. For example, GPT-4o (OpenAI 2024b) demonstrates strong proficiency in solving text-only mathematical problems, but its effectiveness diminishes progressively as the modality tran-

*These authors contributed equally.

sitions from textual to visual. This reduction in capability primarily stems from the current models' inadequate alignment between visual and textual data, which impairs their overall functionality.

To address this issue, we propose Math-PUMA, a methodology centered around **P**rogressive **U**pward **M**ultimodal **A**lignment (PUMA), aimed at enhancing the mathematical reasoning capabilities of MLLMs. Our approach is structured into three distinct stages, with stage 2 serving as the pivotal alignment phase. **(1) Stage 1:** We train the LLM using a substantial dataset of text-based math problems to enhance its problem-solving capabilities. This phase capitalizes on the extensive availability of text-based math problem-solving data. **(2) Stage 2:** It is observed that the model's mathematical problem-solving ability diminished progressively from text to vision, exhibiting an upward pyramidal structure. Consequently, the model's capabilities are categorized into four hierarchical levels. We construct 692K data pairs, with each pair conveying identical information but differing in multimodal representation. By leveraging the KL divergence between next-token prediction distributions for text-rich and vision-rich problems, we achieve progressive bottom-up modal alignment across these hierarchical levels, thereby enhancing the model's ability to tackle multimodal mathematical problems. **(3) Stage 3:** We select 996K high-quality multimodal problem-solving data to fine-tune the model, further enhancing its performance in multimodal mathematical problem-solving tasks.

The contributions of this paper are three-fold:

- We curate a large-scale dataset, Math-PUMA-1M, which comprises 692K data pairs and 996K multimodal mathematical data. This dataset serves as a valuable resource for model training.
- We propose Math-PUMA, a methodology based on Progressive Upward Multimodal Alignment, which enhances mathematical reasoning in MLLMs through a three-stage process.
- Experimental results on three widely-used benchmarks demonstrate that the MLLMs trained with Math-PUMA outperform most open-source models. Notably, our approach effectively narrows the performance gap for problems that contain the same information but are presented in different modalities, as evidenced by results on MATH-VERSE.

Related Work

Multimodal Large Language Models

The exploration of Multimodal Large Language Models (MLLMs) has been inspired by advancements in Large Language Models (LLMs), resulting in remarkable capabilities across a variety of tasks that require both visual and linguistic understanding. CLIP (Radford et al. 2021) is a breakthrough model that learns transferable visual representations from natural language supervision. LLaVA series (Liu et al. 2023, 2024a) pioneer visual instruction tuning for LLMs, employing a simple MLP as a projector to connect the vision encoder with the language model. Models such as Qwen-VL (Bai et al. 2023) and Deepseek-VL (Lu et al. 2024a) introduce a new visual receptor or a hybrid vision encoder, significantly enhancing their ability to perceive and understand visual inputs. However, despite these significant strides, MLLMs still face considerable challenges, particularly in multimodal mathematical reasoning. This is primarily due to the substantial domain gap between the natural scene image and the abstract mathematical graphics. There is a pressing need to enhance the understanding and reasoning abilities of MLLMs in relation to mathematical diagrams.

Multimodal Mathematical Reasoning

The advancement of MLLMs has driven significant research into multimodal reasoning. Current efforts are primarily centered on data augmentation to improve models' performance. Significant efforts have been invested in augmenting text-only mathematical problem-solving data to enhance LLMs' reasoning capabilities (Saxton et al. 2019; Yu et al. 2023; Liu and Yao 2024). G-LLaVA (Gao et al. 2023a) and Math-LLaVA (Shi et al. 2024) improve multimodal mathematical reasoning by constructing the Geo170K and MathV360K datasets, respectively. These are created by generating additional questions for images sourced from public datasets. However, they only serve to expand the text, without increasing the diversity of images in the dataset. GeoGPT4V (Cai et al. 2024) leverages GPT-4V (OpenAI 2023b) to generate new problems and images based on existing datasets, creating a dataset of 4.9K geometric problems combined with 19K open-source data. Nevertheless, due to GPT-4V's subpar capability in generating code from image descriptions, the quality of the generated data is comparatively inferior. By comparison, our work not only makes new advancements in data augmentation, including text rephrasing and the generation of high-quality images, but also introduces a novel alignment method used for training.

Methodology

Data Construction

In order to refine alignment between visual and textual modalities, we need to construct data pairs. We clarify a **"data pair"** as a set of two data components, which share an equivalent level of information within each problem context, and their solutions are identical. A **"data pair"** is defined as two data components that contain equivalent information within the same problem context, and their solutions are identical. However, the distribution of information across different modalities may vary within each pair. We use the term **"vision-rich"** to describe the data component where the visual modality has a higher proportion of information, whereas **"text-rich"** refers to the component with a higher proportion of textual information.

The methods we employ to construct data pairs include automatic data generation and data augmentation based on publicly available sources.

Automatic Data Generation We implement an automatic data generation pipeline for three categories of mathemat-

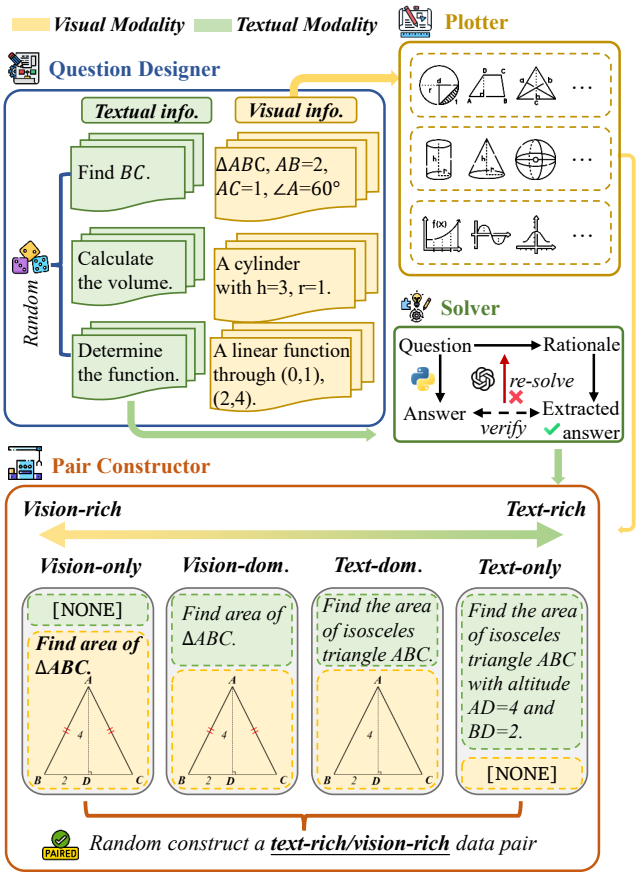

Figure 2: The pipeline of automatic data generation.

ical problems: plane geometry, solid geometry, and functions. The pipeline consists of four agents: **(1) Question Designer**, responsible for formulating problems and assigning information to visual and textual modalities; **(2) Plotter**, which generates diagrams; **(3) Solver**, which provides answers and explanations; and **(4) Pair Constructor**, which produces four types of data and randomly selects two to form a data pair. Figure 2 illustrates this automatic data generation process.

- **Question Design:** The Question Designer employs a random selection process to determine the specific type of mathematical problem to be generated. It also randomly selects the information carrier, deciding whether to present the information as text or an image. This choice dictates the visual information sent to the Plotter and the textual information sent to the Solver.
- **Plotting:** In accordance with the visual information received, the Plotter uses the predefined fundamental tools to plot diagrams.
- **Problem Solving:** The Solver calculates the answer using the text-only version of the problem, which contains complete information. As the calculation is performed programmatically, the answer is reliable. Considering that MLLMs can obtain stronger reasoning abilities from step-by-step solutions, the Solver generates a detailed explanation for each problem by calling GPT-4o-mini (OpenAI 2024a) and verifying the explanations against the standard answer to ensure accuracy.
- **Pair Construction:** The Pair Constructor combines the diagram from the Plotter and the text from the Solver to obtain up to four types of data, each comprising the same information but presented in a different modality: vision-only, vision-dominant, text-dominant, and text-only. Two of these are randomly selected to form a data pair, with the component containing more visual information classified as vision-rich and the other as text-rich.

We generated 40K data each for plane geometry, solid geometry, and functions, summing up to 120K.

Data Augmentation We initially collect 80K mathematical problem-solving data from online sources. By rephrasing the problems from multiple perspectives (Yu et al. 2023) and applying a series of traditional image processing techniques such as scaling, stretching, and gamma transformation, we expand the dataset to 310K. Additionally, we utilize the VisualWebInstruct dataset (TIGER-Lab 2024) containing 262K data. To automate the construction of data pairs, we employ a straightforward text-to-image rendering process to convert the content from textual to visual form. The original data serve as the text-rich component, while the generated data form the vision-rich component. In total, we obtain 572K data pairs.

Training Stages

We employ a three-stage pipeline to train our models, with specific details shown in Figure 3.

Stage 1: Enhancing the Language Model's Mathematical Reasoning Abilities Given the abundance of unsupervised text-based mathematical training corpora and problem-solving data (Shao et al. 2024), in comparison to the scarcity of high-quality multimodal mathematical problem-solving data, we initially train the LLM on a large corpus of text-based math problems to bolster its mathematical reasoning capabilities. To leverage the strengths of existing LLMs that have demonstrated superior performance in mathematical problem-solving (Shao et al. 2024; Yang et al. 2024), we use them to initialize our MLLMs. Subsequently, we fine-tune the model using 200K data extracted from various datasets (Yue et al. 2023; Tong et al. 2024; Mitra et al. 2024; LI et al. 2024). This phase significantly enhances the LLM's mathematical reasoning abilities.

Stage 2: Progressive Upward Multimodal Alignment We observe that the multimodal mathematical reasoning ability of MLLMs resembles a pyramid, with performance declining from bottom to top as the information shifts from text to visual modalities. In order to address the discrepancy in performance between text-rich and vision-rich mathematical reasoning, we propose PUMA (Progressive Upward Multimodal Alignment). The objective of PUMA is to facilitate the effective resolution of vision-rich mathematical problems by aligning the outputs of MLLMs with text-rich data, thereby enhancing their reasoning capabilities across different modalities.

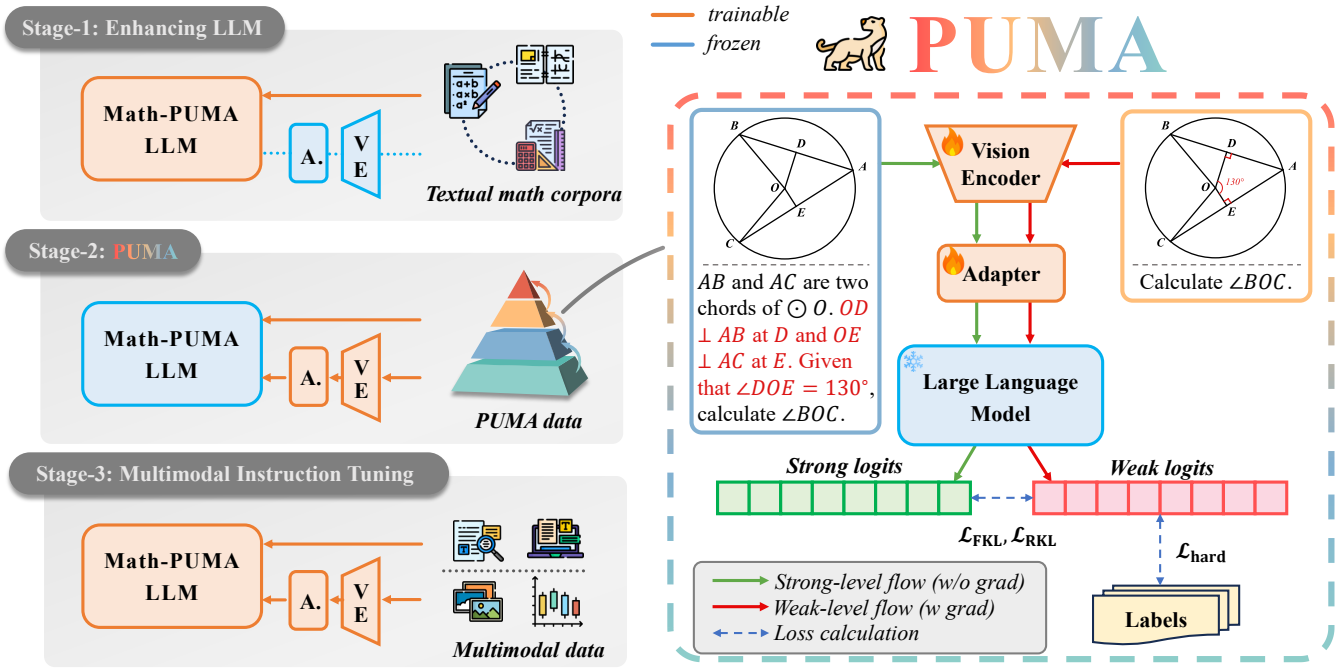

Figure 3: Overview of the Math-PUMA approach. (*Left*) The three stage training process of Math-PUMA. (*Right*) The details for aligning data pair. The input data pair includes text-rich data at the strong level and vision-rich data at the weak level, simultaneously processed by the MLLM. The strong logits and labels are used to supervise the weak logits.

Let $i = 0, 1, 2, 3$ represents the levels of capability for MLLMs, ranging from weak to strong (top-down). For a visual mathematical problem, the inference results of MLLMs are progressively inferior on the i-th level compared to the $(i+1)$-th level. We denote the response distribution (logits) obtained by MLLMs when processing the input of i-th level as p_i, while the response distribution (logits) obtained on the input of $(i+1)$-th level is denoted as p_{i+1}. The forward KL (FKL) divergence and reverse KL (RKL) divergence between these distributions are calculated, since they converge to the same objective after a sufficient number of epochs for MLLMs (Wu et al. 2024).

Let $\mathbf{y}^{(i)} = \{y_t^{(i)}\}_{t=1}^{T}$ represent the response generated by MLLMs based on input $\mathbf{x}^{(i)}$. Here, $y_t^{(i)} \in \{Y_1^{(i)}, Y_2^{(i)}, ..., Y_V^{(i)}\}$, with V representing the vocabulary size. p_i and p_{i+1} represent the distributions of weak and strong levels, $\mathbf{z}^{(i)} = (z_1^{(i)}, z_2^{(i)}, ..., z_V^{(i)})$ and $\mathbf{z}^{(i+1)} = (z_1^{(i+1)}, z_2^{(i+1)}, ..., z_V^{(i+1)})$ represent the logits of weak and strong levels, respectively. The FKL divergence and RKL divergence are computed as follows:

$$\begin{aligned}\mathcal{L}_{\text{FKL}} &= \frac{1}{TV}\sum_{t=1}^{T}\text{KL}\Big(p_i(y_t^{(i)}|\mathbf{y}_{<t}^{(i)})||p_{i+1}(y_t^{(i+1)}|\mathbf{y}_{<t}^{(i+1)})\Big) \\ &= \frac{1}{TV}\sum_{t=1}^{T}\sum_{j=1}^{V} p_i(Y_j^{(i)}|\mathbf{y}_{<t}^{(i)})\log\frac{p_i(Y_j^{(i)}|\mathbf{y}_{<t}^{(i)})}{p_{i+1}(Y_j^{(i+1)}|\mathbf{y}_{<t}^{(i+1)})},\end{aligned} \tag{1}$$

$$\begin{aligned}\mathcal{L}_{\text{RKL}} &= \frac{1}{TV}\sum_{t=1}^{T}\text{KL}\Big(p_{i+1}(y_t^{(i+1)}|\mathbf{y}_{<t}^{(i+1)})||p_i(y_t^{(i)}|\mathbf{y}_{<t}^{(i)})\Big) \\ &= \frac{1}{TV}\sum_{t=1}^{T}\sum_{j=1}^{V} p_{i+1}(Y_j^{(i+1)}|\mathbf{y}_{<t}^{(i+1)})\log\frac{p_{i+1}(Y_j^{(i+1)}|\mathbf{y}_{<t}^{(i+1)})}{p_i(Y_j^{(i)}|\mathbf{y}_{<t}^{(i)})},\end{aligned} \tag{2}$$

with

$$p_i(Y_j^{(i)}|\mathbf{y}_{<t}^{(i)}) = \frac{\exp\left(z_j^{(i)}/\tau\right)}{\sum_{k=1}^{V}\exp(z_k^{(i)}/\tau)}, \tag{3}$$

where τ represents the temperature hyperparameter.

Furthermore, to maintain training stability, we calculate a hard loss by utilizing the solutions of mathematical problems as the ground truth labels, i.e.,

$$\begin{aligned}\mathcal{L}_{\text{hard}} &= -\frac{1}{TV}\sum_{t=1}^{T}\log p_i(y_t^{(i)}|\mathbf{x}^{(i)}, \mathbf{y}_{<t}^{(i)}) \\ &= -\frac{1}{TV}\sum_{t=1}^{T}\sum_{j=1}^{V}\log p_i(Y_j^{(i)}|\mathbf{x}^{(i)}, \mathbf{y}_{<t}^{(i)}).\end{aligned} \tag{4}$$

Finally, the total loss is computed as

$$\mathcal{L} = \lambda_{\text{KL}}(\alpha_{\text{KL}}\mathcal{L}_{\text{FKL}} + (1-\alpha_{\text{KL}})\mathcal{L}_{\text{RKL}})\tau^2 + (1-\lambda_{\text{KL}})\mathcal{L}_{\text{hard}}, \tag{5}$$

where λ_{KL} is a hyperparameter that balances the weight between the combined FKL and RKL divergences and the hard loss term, α_{KL} is a weight hyperparameter that balances the contribution between $\mathcal{L}_{\text{FKL}}$ and $\mathcal{L}_{\text{RKL}}$. The purpose of multiplying KL by τ^2 is to equalize the gradients of the two losses.

At this stage, we use a total of 692K data pairs for training, which includes 120K data pairs automatically generated and 572K data pairs obtained through data augmentation based on publicly available data as described in Data Construction.

Stage 3: Multimodal Instruction Tuning In the final phase, we enhance the model's reasoning capabilities by incorporating multimodal problem-solving data. Initially, we retain the majority of the high-quality data used in Stage 2 and augment our dataset with the MathV360K dataset (Shi et al. 2024). Specifically, we focus on enriching the geometric problem subset within MathV360K, expanding it from 40K to 120K in order to address the scarcity of geometric data. Furthermore, as referenced in (Lu et al. 2024a), we incorporate a balanced amount of textual data to mitigate potential modality imbalances and enhance the model's overall performance. All data include detailed reasoning processes to guide the model's understanding and learning.

Ultimately, we compile a large-scale instruction tuning dataset, comprising a total of 996K data. This multimodal instruction tuning not only bolsters the model's reasoning and problem-solving abilities but also ensures that it can effectively leverage both textual and visual information for improved performance in mathematical problem-solving.

Experiments

Experimental Setup

Models We validate the effectiveness of our method across various base models and scales, including DeepSeek-Math-7B (Shao et al. 2024), Qwen2-1.5B and Qwen2-7B (Yang et al. 2024), chosen as the LLM for Math-PUMA. To ensure the compatibility with DeepSeek-Math and DeepSeek-VL (Lu et al. 2024a), we adhere to the architecture of DeepSeek-VL. For Qwen2, we adopt a similar architecture to LLaVA, with the visual encoder designated as SigLIP-so400m-patch14-384 (Zhai et al. 2023).

Benchmarks We conduct extensive experiments on three popular multimodal mathematical problem-solving benchmarks: MATHVERSE (Zhang et al. 2024a), MATHVISTA (Lu et al. 2024b), and WE-MATH (Qiao et al. 2024). MATHVERSE evaluates the multimodal mathematical reasoning abilities of MLLMs under five different conditions. MATHVISTA comprises samples that require fine-grained, in-depth visual understanding and compositional reasoning, posing a challenge for all baseline models on this benchmark. WE-MATH is the first benchmark specifically designed to explore the problem-solving principles beyond the end-to-end performance.

Evaluation and Metrics We refer to the leaderboards and adopt the official implementations of MATHVERSE, MATHVISTA, and WE-MATH. For MATHVERSE and MATHVISTA, initially, we use GPT-4o-mini (OpenAI 2024a) to extract answers from the responses generated by MLLMs. Subsequently, we employ GPT-4o-mini once more to verify the correctness of the extracted answers. The prompts used for answer extraction and correctness assessment are kept consistent with the official implementation. Ultimately, we calculate the accuracy scores as the evaluation metric. For WE-MATH, we select the average and Rote Memorization (RM) scores as evaluation metrics.

Implementation Details Our experiments are conducted using PyTorch version 2.1.0 and CUDA 12.1, utilizing 32 NVIDIA A100 GPUs with 80GB memory each. The training process is divided into three stages, each with specific hyperparameters and configurations. We employ the AdamW optimizer (Kingma and Ba 2014), configured with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is adjusted across three stages: 3×10^{-5} for stage 1, 5×10^{-5} for stage 2, and 3×10^{-5} for stage 3. A cosine learning rate schedule is implemented with a warm-up phase covering 2% of the total training steps. Additionally, a decay rate of 0.1 is applied. The KL divergence is controlled using specific hyperparameters: α_{KL} is set to 0.2, τ to 1.0, and λ_{KL} to 0.1. The training is conducted over 1 epoch. The batch sizes for three stages are 256, 512, and 256, respectively.

Performance Comparison

Comparison on MATHVERSE MATHVERSE is capable of clearly demonstrating the gap between visual and textual modalities. From Table 1, it can be observed that the MLLMs trained by Math-PUMA achieve the state-of-the-art (SOTA) among open-source MLLMs. Compared to the previous SOTA method, MAVIS, the MLLMs trained by Math-PUMA exhibit accuracy scores improvement about 6%. When compared to the closed-source GPT-4V (OpenAI 2023b), Math-PUMA-Qwen2-7B performs competitively with only a gap of 4.7%, demonstrating the effectiveness of Math-PUMA.

Comparison on MATHVISTA MATHVISTA is a comprehensive benchmark designed to evaluate mathematical reasoning. According to the results presented in Table 1, Math-PUMA-Qwen2-7B demonstrates SOTA performance in GPS, ALG, GEO and SCI domains among open-source MLLMs of the same scale. It outperforms InternLM-XComposer2-VL (Dong et al. 2024) by significant margins, with accuracy improvements of 16.4%, 15.7%, 16.8%, and 4.9% in these respective domains.

Comparison on WE-MATH WE-MATH places strong emphasis on the importance of the mathematical reasoning process. Table 2 demonstrates that Math-PUMA-Qwen2-7B achieves SOTA performance in average scores among open-source MLLMs with approximate 10B parameters, surpassing InternLM-XComposer2-VL. Notably, even among open-source MLLMs with parameters exceeding 20B, Math-PUMA-Qwen2-7B outperforms LLaVA-NeXT (Liu et al. 2024b) 72B model, reaching the performance of LLaVA-NeXT 110B model. While Math-PUMA-Qwen2-7B surpasses Qwen-VL-Max (Bai et al. 2023) among closed-source models, there remains a significant gap compared to GPT-4V and GPT-4o.

Ablation Study

Ablation studies are performed on MATHVERSE to highlight the contribution of each training stage and to assess the im-

Model	# Params.	MATHVERSE						MATHVISTA				
		ALL ↑	Text-dom. ↑	Text-lite ↑	Vision-int. ↑	Vision-dom. ↑	Vision-only ↑	ALL ↑	GPS ↑	ALG ↑	GEO ↑	SCI ↑
Baselines												
Random chance	-	12.4	12.4	12.4	12.4	12.4	12.4	17.9	21.6	21.7	20.1	17.2
Human performance	-	64.9	71.2	70.9	61.4	68.3	66.7	60.3	48.4	50.9	51.4	64.9
Closed-source LLMs												
ChatGPT (Ouyang et al. 2022)	-	-	33.3	18.9	-	-	-	33.2	29.3	31.0	31.0	50.8
GPT-4 (OpenAI 2023a)	-	-	46.5	20.7	-	-	-	33.2	31.7	33.5	32.2	58.2
Closed-source MLLMs												
Qwen-VL-Plus (Bai et al. 2023)	-	11.8	15.7	11.1	9.0	13.0	10.0	43.3	38.5	39.1	39.3	59.0
Gemini-1.0-Pro (Gemini Team 2023)	-	22.3	27.6	23.7	19.4	20.3	20.5	45.2	40.4	45.2	41.0	54.9
Qwen-VL-Max (Bai et al. 2023)	-	24.8	30.3	24.8	20.6	23.3	25.1	-	-	-	-	-
GPT-4V (OpenAI 2023b)	-	**38.3**	**52.1**	**40.9**	**34.9**	**33.6**	**29.8**	**49.9**	**50.5**	**53.0**	**51.0**	**63.1**
Open-source MLLMs												
mPLUG-Owl2 (Ye et al. 2024)	7B	4.6	6.6	6.3	6.3	5.6	4.9	22.2	23.6	23.6	23.9	26.3
LLaMA-Adapter-V2 (Gao et al. 2023b)	7B	5.7	6.2	5.9	6.1	4.2	6.1	23.9	25.5	26.3	24.3	29.5
LLaVA-1.5 (Liu et al. 2024a)	13B	7.6	8.8	7.6	7.4	7.4	6.9	25.7	18.3	19.6	17.6	42.6
LLaVA-NeXT (Liu et al. 2024b)	8B	10.3	12.8	12.0	10.7	9.7	6.3	34.6	-	-	-	-
MiniGPT-v2 (Chen et al. 2023a)	7B	11.0	12.1	12.0	13.1	10.3	7.4	23.1	26.0	28.1	24.7	25.4
SPHINX-Plus (Gao et al. 2024)	13B	12.2	13.9	11.6	11.6	13.5	10.4	36.8	-	-	-	-
ShareGPT4V (Chen et al. 2023b)	13B	13.1	16.2	16.2	15.5	13.8	3.7	27.5	27.4	-	27.6	-
InternLM-XC2. (Dong et al. 2024)	7B	16.3	20.2	14.3	14.2	17.5	15.2	47.8	31.7	32.0	30.5	37.7
G-LLaVA (Gao et al. 2023a)	7B	16.6	20.9	20.7	17.2	14.6	9.4	23.8	38.9	36.3	35.6	20.5
SPHINX-MoE (Gao et al. 2024)	8×7B	16.8	26.2	17.4	16.7	12.5	11.1	42.3	31.2	31.7	30.5	**50.8**
DeepSeek-VL (Lu et al. 2024a)	7B	19.3	23.0	23.2	20.2	18.4	11.8	34.9	28.4	29.2	27.2	35.3
Math-LLaVA (Shi et al. 2024)	13B	22.9	27.3	24.9	24.5	21.7	16.1	38.3	29.3	28.5	30.5	42.6
MAVIS (Zhang et al. 2024b)	7B	27.5	41.4	29.1	27.4	24.9	14.6	-	-	-	-	-
Math-PUMA-Qwen2-1.5B	1.5B	29.6	35.8	32.2	31.3	30.4	18.5	44.5	47.6	43.4	**47.3**	41.0
Math-PUMA-Qwen2-7B	7B	**33.6**	42.1	35.0	33.4	**31.6**	**26.0**	**47.9**	**48.1**	**47.7**	**47.3**	42.6
Math-PUMA-DeepSeek-Math-7B	7B	31.8	**43.4**	**35.4**	**33.6**	**31.6**	14.7	44.7	39.9	39.2	41.4	48.4

Table 1: **Mathematical evaluation on MATHVERSE and MATHVISTA *testmini* sets.** For MATHVERSE, we calculate the "ALL" score without averaging the "Text-only" version. For MATHVISTA, we select 4 mathematical categories from the original 12 categories. ALL: overall accuracy across original categories; GPS: geometry problem solving; ALG: algebraic reasoning; GEO: geometry reasoning; SCI: scientific reasoning. For closed-source and open-source MLLMs, the best accuracy scores are marked in **bold** fonts, while the second best accuracy scores are marked in underline fonts, respectively.

pact of their sequential order on Math-PUMA.

The Role of Each Stage To evaluate the significance of each stage, we conduct three ablation experiments by individually removing stages 1, 2, and 3. We then assess the accuracy across overall, text-dominant, and vision-only scenarios, as well as the gaps between them. The results of these experiments are summarized in Table 3.

Removing Stage 1: Stage 1 aims to enhance the mathematical reasoning capabilities of the LLMs. As observed in Table 3, upon removing stage 1, there is a slight decrease in the accuracy compared to the corresponding model trained with all three stages. This reduction occurs because stage 1 serves as the foundation for stage 2. When the LLM lacks strong mathematical reasoning capabilities, strong logits are not reliable to supervise weak logits, resulting in lower performance. However, due to the presence of complete stage 2 and 3, the gap remains close to that of the complete three-stage training model and relatively low.

Removing Stage 2: Stage 2 embodies our devised PUMA, facilitating a close alignment between visual and textual modalities. As depicted in Table 3, the absence of stage 2 results in a wider gap in reasoning performance between textual and visual modalities when compared to the three-stage approach. Nonetheless, with the enhancement of mathematical reasoning capabilities by stage 1 and multimodal instruction tuning with high-quality data through stage 3, the overall performance persists at a high level.

Removing Stage 3: Stage 3 is multimodal instruction tuning. We observe that if only stage 1 and 2 are performed without subsequent multimodal instruction tuning, MLLMs tend to lose instruction-following (IF) capabilities to some extent. As seen in Table 3, the performance of MLLMs drastically declines when stage 3 is excluded, primarily due to the loss of IF capabilities. Since we have conducted stage 2, the gap between textual and visual modalities remains relatively small.

Sequential Order of Stages We swap stage 2 and 3 to assess their impact on MLLMs. As shown in Table 3, exchanging stage 2 and 3 leads to a significant performance drop. Our analysis of each stage reveals the critical role of stage 3 in maintaining the IF capabilities of MLLMs. Consequently, rearranging the stage 2 and 3 results in the loss of IF capabilities of MLLMs, thereby influencing their overall performance. Nonetheless, the eventual implementation of stage 2 ensures that the gap between textual and visual modalities remains relatively small.

Model	# Params.	AVG ↑	RM ↓
Closed-source MLLMs			
Qwen-VL-Max (Bai et al. 2023)	-	10.5	75.5
Gemini-1.5-Pro (Reid et al. 2024)	-	26.4	54.8
GPT-4V (OpenAI 2023b)	-	31.1	47.9
GPT-4o (OpenAI 2024b)	-	**42.9**	**34.2**
Open-source MLLMs (≥20B)			
InternVL-Chat-V1.5 (Chen et al. 2024)	26B	15.0	73.3
LLaVA-NeXT (Liu et al. 2024b)	72B	13.4	71.0
LLaVA-NeXT (Liu et al. 2024b)	110B	**19.2**	**66.0**
Open-source MLLMs (≈10B)			
LLaVA-1.5 (Liu et al. 2024a)	7B	6.5	85.6
LLaVA-1.5 (Liu et al. 2024a)	13B	8.4	78.1
LLaVA-1.6 (Liu et al. 2024b)	7B	3.3	89.1
LLaVA-1.6 (Liu et al. 2024b)	13B	5.2	86.9
DeepSeek-VL (Lu et al. 2024a)	7B	6.3	84.8
G-LLaVA (Gao et al. 2023a)	13B	6.5	86.6
Math-LLaVA (Shi et al. 2024)	13B	11.1	72.8
InternLM-XC2. (Dong et al. 2024)	7B	12.7	77.6
Math-PUMA-Qwen2-1.5B	1.5B	10.4	75.5
Math-PUMA-Qwen2-7B	7B	**19.2**	67.8
Math-PUMA-DeepSeek-Math	7B	15.6	**67.4**

Table 2: **Evaluation results on WE-MATH *testmini* set.** AVG: average score (strict); RM: rote memorization (strict). The best scores of each category are marked in **bold** fonts.

Order	LLM	ALL ↑	Text-dom. ↑	Vision-only ↑	Gap ↓
Standard pipeline					
1 → 2 → 3	Qwen2-1.5B	**29.6**	35.8	**18.5**	93.5
	Qwen2-7B	**33.6**	42.1	**26.0**	**61.9**
	DeepSeek-Math	**31.8**	**43.4**	14.7	195.2
Effectiveness of Stage 1 (Enhancing LLM)					
2 → 3	Qwen2-1.5B	17.0	19.9	12.1	**64.5**
	Qwen2-7B	19.6	27.3	11.9	129.4
	DeepSeek-Math	23.9	30.7	11.2	174.1
Effectiveness of Stage 2 (Math-PUMA)					
1 → 3	Qwen2-1.5B	24.6	**40.3**	9.8	311.2
	Qwen2-7B	27.2	**44.1**	11.0	300.9
	DeepSeek-Math	29.3	**43.4**	9.1	376.9
Effectiveness of Stage 3 (Multimodal instruction tuning)					
1 → 2	Qwen2-1.5B	11.7	15.5	8.1	91.4
	Qwen2-7B	21.2	28.9	12.2	136.9
	DeepSeek-Math	22.2	36.2	**14.8**	**144.6**
Sequential Order of Stages					
1 → 3 → 2	Qwen2-1.5B	24.5	38.2	12.1	215.7
	Qwen2-7B	26.7	34.4	18.7	84.0
	DeepSeek-Math	23.4	34.3	4.3	697.7

Table 3: **Results of ablation study.** Order: the sequential order of Stage 1, 2, and 3; ALL: overall accuracy; Text-dom.: accuracy of text-dominant data; Vision-only: accuracy of vision-only data; Gap: (Text-dom. − Vision-only) / Vision-only. The best scores of each LLM are marked in **bold** fonts.

Have the modality gaps truly narrowed? Through the aforementioned analysis, we have demonstrated the effectiveness of our method. However, we still seek to provide a definitive conclusion to address the initial query: Has the performance gap between different modalities truly narrowed? To this end, we base our exploration on the evaluation metrics provided by MATHVERSE, calculating the average scores of the model on textual and visual questions to intuitively assess the model's performance across these two distinct modalities. Additionally, we compute the skewness and coefficient of variation of the scores on different types of questions in MATHVERSE to corroborate our observations regarding the modal balance.

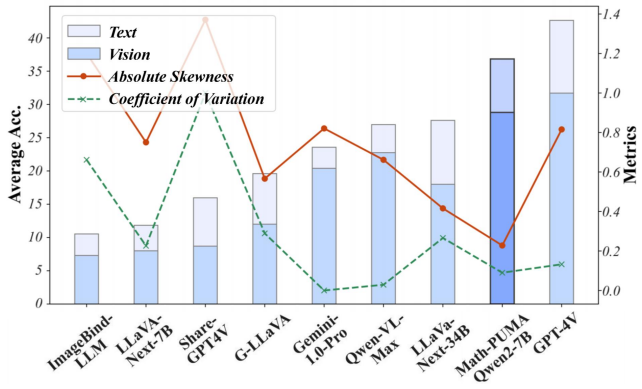

Figure 4: Visualizing MLLMs' performance on MATHVERSE. "Text" shows average scores for text-dominant and text-lite categories, while "Vision" represents average scores for vision-intensive, vision-dominant, and vision-only categories. "Absolute Skewness" and "Coefficient of Variance" denote the statistical measures of score distribution across the five categories, with skewness taken as an absolute value.

As illustrated in Figure 4, in terms of overall performance, our model achieves high average scores on both textual and visual questions, outperforming closed-source MLLMs such as Gemini-1.0-Pro and Qwen-VL-Max. We analyze the performance gap between textual and visual modalities. Our model maintains a high level of performance while exhibiting a relatively smaller gap, which is even smaller than that of GPT-4V. Additionally, regarding score distribution, a model that performs consistently across modalities should demonstrate similar scores across various types of questions in MATHVERSE. This consistency is indicated by lower absolute skewness and coefficient of variation. By visualizing the score distributions of several models, it is evident that our model exhibits low levels of both skewness and coefficient of variation, indicating a well-balanced performance across different types. In summary, our method mitigates the performance disparity between different modalities.

Conclusion

In this paper, we present Math-PUMA, a progressive upward multimodal alignment approach aimed at enhancing the mathematical reasoning capabilities of MLLMs. Experimental results indicate that Math-PUMA MLLMs not only achieve state-of-the-art performance among open-source models on multiple mathematical benchmarks but also significantly reduce the performance gap between textual and visual modalities.

References

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-VL: A Frontier Large Vision-Language Model with Versatile Abilities. *arXiv preprint arXiv:2308.12966*.

Cai, S.; Bao, K.; Guo, H.; Zhang, J.; Song, J.; and Zheng, B. 2024. GeoGPT4V: Towards Geometric Multi-modal Large Language Models with Geometric Image Generation. *arXiv preprint arXiv:2406.11503*.

Chen, J.; Li, D. Z. X. S. X.; Zhang, Z. L. P.; Xiong, R. K. V. C. Y.; and Elhoseiny, M. 2023a. MiniGPT-V2: Large Language Model as a Unified Interface for Vision-Language Multi-task Learning. *arXiv preprint arXiv:2310.09478*.

Chen, L.; Li, J.; wen Dong, X.; Zhang, P.; He, C.; Wang, J.; Zhao, F.; and Lin, D. 2023b. ShareGPT4V: Improving Large Multi-Modal Models with Better Captions. *ArXiv*, abs/2311.12793.

Chen, W.; Ma, X.; Wang, X.; and Cohen, W. W. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Chen, Z.; Wang, W.; Tian, H.; Ye, S.; Gao, Z.; Cui, E.; Tong, W.; Hu, K.; Luo, J.; Ma, Z.; et al. 2024. How Far Are We to GPT-4V? Closing the Gap to Commercial Multimodal Models with Open-Source Suites. *arXiv preprint arXiv:2404.16821*.

Dong, X.; Zhang, P.; Zang, Y.; Cao, Y.; Wang, B.; Ouyang, L.; Wei, X.; Zhang, S.; Duan, H.; Cao, M.; et al. 2024. InternLM-XComposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*.

Gao, J.; Pi, R.; Zhang, J.; Ye, J.; Zhong, W.; Wang, Y.; Hong, L.; Han, J.; Xu, H.; Li, Z.; et al. 2023a. G-LLaVA: Solving Geometric Problem with Multi-Modal Large Language Model. *arXiv preprint arXiv:2312.11370*.

Gao, P.; Han, J.; Zhang, R.; Lin, Z.; Geng, S.; Zhou, A.; Zhang, W.; Lu, P.; He, C.; Yue, X.; Li, H.; and Qiao, Y. 2023b. LLaMA-Adapter V2: Parameter-Efficient Visual Instruction Model. *arXiv preprint arXiv:2304.15010*.

Gao, P.; Zhang, R.; Liu, C.; Qiu, L.; Huang, S.; Lin, W.; Zhao, S.; Geng, S.; Lin, Z.; Jin, P.; et al. 2024. SPHINX-X: Scaling Data and Parameters for a Family of Multi-modal Large Language Models. *arXiv preprint arXiv:2402.05935*.

Gemini Team, G. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gou, Z.; Shao, Z.; Gong, Y.; Yang, Y.; Huang, M.; Duan, N.; Chen, W.; et al. 2023. Tora: A tool-integrated reasoning agent for mathematical problem solving. *arXiv preprint arXiv:2309.17452*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

LI, J.; Beeching, E.; Tunstall, L.; Lipkin, B.; Soletskyi, R.; Huang, S. C.; Rasul, K.; Yu, L.; Jiang, A.; Shen, Z.; Qin, Z.; Dong, B.; Zhou, L.; Fleureau, Y.; Lample, G.; and Polu, S. 2024. NuminaMath. https://huggingface.co/AI-MO/NuminaMath-CoT.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, 12888–12900. PMLR.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26296–26306.

Liu, H.; Li, C.; Li, Y.; Li, B.; Zhang, Y.; Shen, S.; and Lee, Y. J. 2024b. LLaVA-NeXT: Improved Reasoning, OCR, and World Knowledge.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual Instruction Tuning. In *NeurIPS*.

Liu, H.; and Yao, A. C.-C. 2024. Augmenting math word problems via iterative question composing. *arXiv preprint arXiv:2401.09003*.

Lu, H.; Liu, W.; Zhang, B.; Wang, B.; Dong, K.; Liu, B.; Sun, J.; Ren, T.; Li, Z.; Sun, Y.; et al. 2024a. Deepseek-VL: Towards Real-world Vision-Language Understanding. *arXiv preprint arXiv:2403.05525*.

Lu, P.; Bansal, H.; Xia, T.; Liu, J.; Li, C.; Hajishirzi, H.; Cheng, H.; Chang, K.-W.; Galley, M.; and Gao, J. 2024b. MathVista: Evaluating Mathematical Reasoning of Foundation Models in Visual Contexts. In *The Twelfth International Conference on Learning Representations*.

Mitra, A.; Khanpour, H.; Rosset, C.; and Awadallah, A. 2024. Orca-math: Unlocking the potential of slms in grade school math. *arXiv preprint arXiv:2402.14830*.

OpenAI. 2023a. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2023b. GPT-4V(ision) System Card. https://openai.com/index/gpt-4v-system-card/. Accessed: 2025-01-27.

OpenAI. 2024a. GPT-4o mini: advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence. Accessed: 2025-01-27.

OpenAI. 2024b. GPT-4o System Card. https://openai.com/index/gpt-4o-system-card/. Accessed: 2025-01-27.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Gray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Qiao, R.; Tan, Q.; Dong, G.; Wu, M.; Sun, C.; Song, X.; GongQue, Z.; Lei, S.; Wei, Z.; Zhang, M.; et al. 2024. We-Math: Does Your Large Multimodal Model Achieve Human-like Mathematical Reasoning? *arXiv preprint arXiv:2407.01284*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Reid, M.; Savinov, N.; Teplyashin, D.; Lepikhin, D.; Lillicrap, T.; Alayrac, J.-b.; Soricut, R.; Lazaridou, A.; Firat, O.; Schrittwieser, J.; et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Saxton, D.; Grefenstette, E.; Hill, F.; and Kohli, P. 2019. Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv:1904.01557*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Zhang, M.; Li, Y.; Wu, Y.; and Guo, D. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*.

Shi, W.; Hu, Z.; Bin, Y.; Liu, J.; Yang, Y.; Ng, S.-K.; Bing, L.; and Lee, R. K.-W. 2024. Math-LLaVA: Bootstrapping Mathematical Reasoning for Multimodal Large Language Models. *arXiv preprint arXiv:2406.17294*.

TIGER-Lab. 2024. VisualWebInstruct.

Tong, Y.; Zhang, X.; Wang, R.; Wu, R.; and He, J. 2024. DART-Math: Difficulty-Aware Rejection Tuning for Mathematical Problem-Solving. *arXiv preprint arXiv:2407.13690*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wu, T.; Tao, C.; Wang, J.; Zhao, Z.; and Wong, N. 2024. Rethinking Kullback-Leibler Divergence in Knowledge Distillation for Large Language Models. *arXiv preprint arXiv:2404.02657*.

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

Ye, Q.; Xu, H.; Ye, J.; Yan, M.; Hu, A.; Liu, H.; Qian, Q.; Zhang, J.; and Huang, F. 2024. mPLUG-Owl2: Revolutionizing Multi-modal Large Language Model with Modality Collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13040–13051.

Yu, L.; Jiang, W.; Shi, H.; Yu, J.; Liu, Z.; Zhang, Y.; Kwok, J. T.; Li, Z.; Weller, A.; and Liu, W. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Yue, X.; Qu, X.; Zhang, G.; Fu, Y.; Huang, W.; Sun, H.; Su, Y.; and Chen, W. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Zhai, X.; Mustafa, B.; Kolesnikov, A.; and Beyer, L. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11975–11986.

Zhang, R.; Jiang, D.; Zhang, Y.; Lin, H.; Guo, Z.; Qiu, P.; Zhou, A.; Lu, P.; Chang, K.-W.; Gao, P.; et al. 2024a. MathVerse: Does Your Multi-modal LLM Truly See the Diagrams in Visual Math Problems? *arXiv preprint arXiv:2403.14624*.

Zhang, R.; Wei, X.; Jiang, D.; Zhang, Y.; Guo, Z.; Tong, C.; Liu, J.; Zhou, A.; Wei, B.; Zhang, S.; et al. 2024b. MAVIS: Mathematical Visual Instruction Tuning. *arXiv preprint arXiv:2407.08739*.