ChatGLM-Math: Improving Math Problem-Solving in Large Language Models with a Self-Critique Pipeline

Yifan Xu[1†], Xiao Liu[12†], Xinghan Liu[12], Zhenyu Hou[12], Yueyan Li[23], Xiaohan Zhang[2], Zihan Wang[12], Aohan Zeng[12], Zhengxiao Du[12], Wenyi Zhao[2], Jie Tang[2*], Yuxiao Dong[2*]

[1]Tsinghua University [2]Zhipu.AI [3]Beijing U. of Posts and Telecoms

Abstract

Large language models (LLMs) have shown excellent mastering of human language but still struggle in real-world applications that require mathematical problem-solving. While many strategies and datasets to enhance LLMs' mathematics are developed, it remains a challenge to simultaneously maintain and improve both language and mathematical capabilities in deployed LLM systems. In this work, we tailor the Self-Critique pipeline, which addresses the challenge in the feedback learning stage of LLM alignment. We first train a general Math-Critique model from the LLM itself to provide feedback signals. Then, we sequentially employ rejective fine-tuning and direct preference optimization over the LLM's own generations for data collection. Based on ChatGLM3-32B, we conduct experiments on both academic and our newly created challenging dataset, MATH-USEREVAL. Results show that our pipeline significantly enhances the LLM's mathematical problem-solving while still improving its language ability, outperforming LLMs that could be two times larger. Related techniques have been deployed to ChatGLM, an online serving LLM. Related evaluation datasets and scripts are released at `https://github.com/THUDM/ChatGLM-Math`.

1 Introduction

Large Language Models (LLMs) (Brown et al., 2020; Chowdhery et al., 2022; Kaplan et al., 2020; Scao et al., 2022; Touvron et al., 2023a; Zeng et al., 2022; Anthropic, 2023) have garnered widespread attention for their remarkable proficiency in various linguistic tasks such as text summarization(Hermann et al., 2015; Völske et al., 2017; Narayan et al., 2018; Li et al., 2022), question answering (Hendrycks et al., 2021a; Kwiatkowski et al., 2019; Bisk et al., 2020), and role-playing conversations (Tu et al., 2024; Zhou et al., 2023a; Shao et al., 2023). Furthermore, their potential in addressing complex problems requiring mathematical reasoning (Yu et al., 2023; Wang et al., 2023; Luo et al., 2023) has expanded their applicability across real-world missions (Liu et al., 2023b; Bai et al., 2023b).

Model	Avg. of GSM8k & MATH	AlignBench Language
DeepSeek-67B-Chat (DeepSeek-AI et al., 2024)	58.3	7.11
DeepSeek-67B-Chat-DPO (DeepSeek-AI et al., 2024)	57.7 (-1.2%)	7.60 (+6.8%)
InternLM2-Chat-20B (Team, 2023)	57.2	7.68
Math-InternLM2-20B (Team, 2023)	60.2 (+5.1%)	6.53 (-14.8%)
ChatGLM3-32B-SFT-2312	52.4	7.37
+ RFT&DPO	61.6 (+17.5%)	7.80 (+5.85%)

Table 1: Our self-critique pipeline enables simultaneous improvement of language and mathematical abilities. Previous alignment methods enhance language but could potentially impair mathematical abilities (DeepSeek-AI et al., 2024), whereas math-specialized models could harm language capabilities (Team, 2023).

Despite these advances, optimizing LLMs to excel simultaneously in language understanding and mathematical problem-solving presents a notable challenge. The prevalent reinforcement learning from human feedback (RLHF) approach primarily enhances text generation based on reward models reflecting human preferences (Touvron et al., 2023a; Ouyang et al., 2022; Touvron et al., 2023b). Although this method boosts the quality of the generated text, it often overlooks the accuracy and logical coherence essential for solving mathematical problems, leading to a discrepancy in performance known as the "alignment tax"(Askell et al., 2021) when applied to mathematical reasoning (refer to Table 1). Conversely, attempts to bolster LLMs' mathematical capabilities typically entail supervised fine-tuning (SFT) that inadvertently diminishes their linguistic versatility, posing a dilemma

[†]Yifan and Xiao contributed equally. Emails: xu-yf23@mails.tsinghua.edu.cn,shawliu9@gmail.com
[*]Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 9733–9760
November 12-16, 2024

for practical applications of LLM systems (Team, 2023; Yu et al., 2023; Luo et al., 2023; Yue et al., 2023).

Pipeline: Self-Critique. This paper introduces a novel approach aimed at enhancing LLMs' linguistic and mathematical skills without compromising one for the other. Our strategy deviates from traditional RLHF by incorporating a Math-Critique model derived from the LLM, which evaluates its mathematical outputs. This self-critique mechanism enables the model to learn from AI-generated feedback specifically tailored to mathematical content (Bai et al., 2022; Lee et al., 2023). Our methodology comprises two primary phases:

- **Stage 1: Rejective Fine-tuning (RFT)** (Yuan et al., 2023) employs a rejection sampling technique, wherein responses failing to meet Math-Critique standards are discarded, while the rest undergo further fine-tuning. This stage aims to enhance the model's accuracy and consistency in mathematical responses while ensuring diversity among the selected answers.
- **Stage 2: Direct Preference Optimization (DPO)** (Rafailov et al., 2023) extends the improvement process by directly learning from pairs of correct and incorrect answers, further refined through Math-Critique, focusing on the most challenging questions from the previous stage.

Benchmark: MATHUSEREVAL. To accurately assess LLMs' capabilities in solving real-world mathematical problems, we develop the MATHUSEREVAL dataset. It features a diverse range of questions, extending beyond academic exercises to include practical application scenarios, thereby better-reflecting user needs compared to traditional academic math datasets (Zhao et al., 2020; Wang et al., 2017; Cobbe et al., 2021). We leverage both GPT-4-turbo and our Math-Critique model for comprehensive scoring.

In summary, our contributions include:

- The introduction of the Self-Critique pipeline, a novel framework that elevates both the mathematical and linguistic capabilities of LLMs through self-generated feedback, thereby eliminating the need for external supervisory models and manual annotations. This approach has been validated on a ChatGLM3-32B model, achieving unparalleled performance on the MATHUSEREVAL, Ape210k (Zhao et al., 2020), MATH (Hendrycks et al., 2021a), and the linguistic tasks of AlignBench (Liu et al., 2023a).
- The creation of the MATHUSEREVAL benchmark, tailored to assess LLMs on complex, open-ended mathematical queries relevant to real-world applications, setting a new standard in evaluating practical mathematical reasoning capabilities.
- A detailed analysis of the key factors contributing to enhancing mathematical proficiency through the Self-Critique pipeline, offering insights into future directions for autonomous model improvement.

2 Related Work

LLM for Math Problem-Solving. Various approaches have been explored to enhance the mathematical problem-solving abilities of language models. Prompting Methods, initiated by Chain of Thought prompting (Wei et al., 2023a; Cheng et al., 2023), have been refined for detailed reasoning, with enhancements from (Yao et al., 2023; Besta et al., 2023; Yang et al., 2023a). Supervised Fine-tuning and Reinforcement Learning (RL) are also pivotal, with high-quality supervisory data from works like (Luo et al., 2023; Yuan et al., 2023; Chern et al., 2023; Yu et al., 2023; Yue et al., 2023; Zhang et al., 2024) directly improving capabilities. RL's potential in general domains is shown by (OpenAI, 2023; Touvron et al., 2023a; DeepSeek-AI et al., 2024; Lightman et al., 2023; Luo et al., 2023; Wang et al., 2023), despite challenges in applying the DPO algorithm (Rafailov et al., 2023) for mathematical tasks. For a detailed comparison with similar works, refer to Table 2.

Mathematical Evaluation. Complex reasoning tasks, such as mathematics, are key indicators of language model capabilities (Koncel-Kedziorski et al., 2016; Polu and Sutskever, 2020; Hendrycks et al., 2021b; Fu et al., 2023). The GSM8k (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b) datasets are widely used benchmarks. Some sets (Ling et al., 2017; Zhong et al., 2023) focus on pure math prowess, while others (Mishra et al., 2022; Suzgun et al., 2022) combine math with other abilities. In Chinese, CM17k (Qin et al., 2021), CARP (Zhang et al., 2023a), Math23K (Wang et al., 2017) and CMath (Wei et al., 2023b) target elementary and middle school math, while AgiEval (Zhong et al., 2023) and

Table 2: Compare ChatGLM-Math with other works with General or Math improvement. We are the first to generate and critique training responses without human annotators and external LLMs.

	Domain	SFT Data Selection	RL Reward	RL Method	w/o Human	w/o Ex. LLM	Self-Gen. Training	Self-Cri.
Instruct GPT	General	-	RM	PPO		✓		
Self-Instruct	General	By Rule	-	-	✓	✓	✓	
Alpagasus	General	By ChatGPT	-	-	✓		✓	
RLAIF	General	RM	RM	PPO	✓			
SPIN	General	-	-	DPO-like		✓		
Metamath	Math	Answer	-	-	✓			
WizardMath	Math	Answer	IRM+PRM	PPO	✓		✓	
ChatGLM-Math	Math & General	Math-Critique	Math-Critique	DPO	✓	✓	✓	✓

Table 3: Compare MathUserEval with other Math benchmarks. "MC" refers to multiple-choice.

Benchmarks	Data Source	Language	Domain				Multiform Answer	Eval. Metric
		Chinese	pre-High School	High School	Col-ledge	Cross Subj.		
Ape210k	annotators	✓	✓					EM
Cmath	books & exams	✓	✓					EM
CM17K	books & exams	✓	✓	✓				EM
CARP	books & exams	✓	✓	✓				EM
TAL-SCQ5K	unknown	✓	✓	✓				MC Acc.
GSM8k	annotators		✓					EM
MATH	math competition			✓	✓		✓	EM
MathUserEval	real-user scenarios & exams	✓	✓	✓	✓	✓	✓	Model Judge (w/ CoT)

GaoKaoBench (Zhang et al., 2023b) present exam-level challenges. However, these datasets are in fixed formats, and simple perturbations can significantly impact performance (Kumar et al., 2021; Zhou et al., 2023b). Thus, performance on these datasets must reflect real-world user questions. For detailed comparisons with similar benchmarks, refer to Table 3.

3 Math-Critique: A General Critic for Math

Definition. We propose Math-Critique, an evaluation model inspired by large models used for assessment (Ke et al., 2023; Zheng et al., 2023). It scores mathematical responses based on questions and reference answers, providing explanatory analysis and a score from 1 to 10. Unlike traditional reward models, Math-Critique enhances judgment accuracy by incorporating reference answers and explanatory analysis inspired by thought chains. Responses are categorized into four types: entirely incorrect (1–2), partially correct methodology with errors (3–5), accurate conclusion with flawed methodology (6–8), and wholly correct (9–10). Math-Critique can be defined as:

$$\mathtt{MC}(\mathrm{Q}, \mathrm{R}, \mathrm{A}) \rightarrow (\mathrm{Critique}, \mathrm{Score})$$

Q is the question, R is the reference answer, and A is the evaluated answer. We used two evaluation methods: average score evaluation, computing the mean critique scores, and hard-split evaluation, classifying answers as passing or failing based on a correctness threshold and then calculating the proportion of correct answers.

Data Collection. Our construction method involves the following steps:

- We designed the scoring rules and intervals for mathematical responses.
- We filtered a dataset from the training data, including questions, reference answers, and model responses. We utilized model sampling answers from multiple sources, including different versions of ChatGLM and other models.
- We employed CritiqueLLM (Ke et al., 2023) to annotate the dataset, selecting annotations that represented the best and worst scoring extremes from these models, and directly used these **pseudo tags** for training. This step generated a total of 10k annotated data entries.
- For results with scores in the middle range, we selected a portion for **manual annotation** into four categories and then mapped these outcomes to a 10-point scale, generating 5k annotated data. We also divided an 800-sample test set into the same method.

4 The Self-Critique Pipeline

Overview. Based on the construction method of Math-Critique, this section introduces the **Self-Critique** pipeline. This pipeline is a weakly supervised iterative training method for enhancing mathematical abilities originating from a single model. Initially, we train a Math-Critique model using the base model and concurrently train a basic Chat Model using the fundamental SFT dataset. Subsequently, we employ the Math-Critique model to supervise the fine-tuning of the Chat Model through rejection sampling. The outcome of this step can serve as a new base model to update both the Math-Critique model and the rejection sampling supervised fine-tuning model. Building upon these steps, our final action involves utilizing the latest Math-Critique model to sample contrast data and then proceeding with DPO training. In the following formula, we use `MC` to represent `MathCritique`.

In these steps, the data construction for the Math-critique-base involves a small amount of manual annotation. However, this batch of annotations is a one-time effort, as only this batch of annotated data

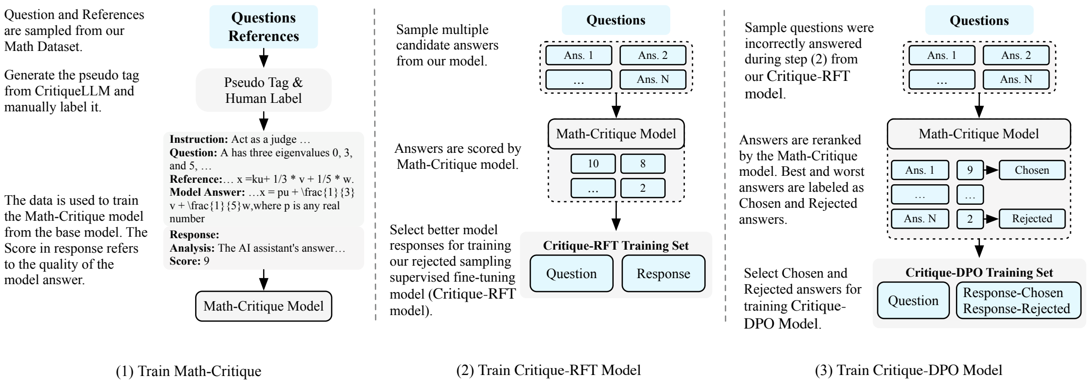


Figure 1: Self-Critique pipeline for ChatGLM-Math. It comprises three steps: training the Math-Critique model, utilizing Math-Critique judgements for sampling, followed by two stages of training: Critique RFT and Critique DPO. Throughout the entire process, only a minimal amount of manual involvement is required during the Math-Critique training phase. Subsequent steps can be fully automated and do not depend on external supervisory models.

is needed as a bootstrap for the remaining iterations. After that, inference and automatic model filtering can complete all remaining steps. Replacing manual annotation with inference can significantly reduce the time required for each iteration from the base model to the final chat model.

4.1 Stage 1: Rejective Fine-tuning

We utilized a rejection sampling method based on Math-Critique. We found that both the sampling range and the model influence the outcomes during the rejection sampling process. Specifically, we designed the following sampling principles:

- Pre-deduplication: Cluster question embeddings from the training set and evenly sample across categories, ensuring a diverse range of questions without repetition.
- Post-sampling deduplication: We conducted a selection process after 5-10 sampling iterations based on the results from Math-Critique. After essential deduplication, we chose the responses only in cases where there were correct and incorrect responses to the same question.

Following the process outlined above, we have obtained the Critique-RFT dataset:

$$D_{\text{RFT}} = \left\{ (q_i, a_{ij}) \mid \begin{array}{l} \frac{1}{n}\sum_x \texttt{MC}(a_{ix}) < 1 \\ \text{and } \texttt{MC}(a_{ij}) > \text{correct-bound} \end{array} \right\}$$

In this dataset, q_i denotes the ith sampled question, with each question undergoing n samplings. a_{ij} represents the jth response to the ith question. MC refers to Math-Critique score. 'correct bound' denotes the minimum acceptable score for a correct answer, generally set at 0.7.

4.2 Stage 2: Direct Preference Optimization

We employed the DPO method to enhance model capabilities after Critique RFT. The primary advantages are its simplicity in constructing data flows, stability, and speed during training. The DPO method directly compares the correct and incorrect answers to the same question. In our approach, both answers are sampled from the model post-RFT, which we found to be critically important.

Our DPO data filtering process is similar to Critique RFT, with the sole difference being the construction method of DPO training pairs. For the selection of DPO pairs, under the premise that there is at least one correct and one incorrect answer, we choose the data pair with the most significant difference in Math-Critique scoring results.

Following the process outlined above, we have obtained the Critique-DPO dataset:

$$D_{\text{DPO}} = \left\{ (q_i, a_{\text{c}}, a_{\text{r}}) \middle| \begin{array}{l} \frac{1}{n}\sum_x \texttt{MC}(a_{ix}) < 1, \\ \texttt{MC}(a_{\text{c}}) > \text{cor.-bound}, \\ \texttt{MC}(a_{\text{r}}) < \text{rej.-bound} \end{array} \right\}$$

In this dataset, each element is a tuple, where q_i is the ith sampled question. For every question q_i, sampled n responses, each denoted by a_{ix}. The Math-Critique (MC) score is computed for each response a_{ix}, and the average of these scores must be less than 1. The chosen answer for each question, a_c, is the one that exceeds the 'correct-bound', which is a predetermined threshold indicating a satisfactory level of correctness, often set above a specific value. Conversely, a_r represents the answer that falls below the 'rejected-bound', which is the threshold below which answers are considered incorrect or unsatisfactory.

4.3 Training

Math-Critique Training We employ the base model of ChatGLM3-32B (Zeng et al., 2022; Du et al., 2022) as the initial Math-Critique base model. After each iteration, the model currently refined through SFT (Supervised Finetuning) or Critique RFT will be used as the base. We use a learning rate 3e-6 and a batch size 128.

Critique-RFT Training During the Critique RFT phase, each of our finetuning iterations includes the datasets from previous stages after deduplication, which also encompasses the initial sft dataset. We merge D_{RFT} and D_{SFT} in this phase. The D_{SFT} dataset encompasses many routine tasks and can be substituted with an open-source instruction finetuning dataset. To eliminate the potential interference of this dataset on the final results, we compared the impact of including or excluding the sft data in our ablation study. We finetune a base LLM model π_θ by standard max-loglikelihood loss. In this stage, we use a learning rate 2e-5 and finetune for 8000 steps with a batch size of 64.

Critique-DPO Training During the Critique-DPO phase, it was observed that the direct use of DPO loss led to instability in the training process. To mitigate this issue, a cross-entropy loss for the chosen answer was introduced as a regularization term to the total loss. The loss function we used is:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(q_i, a_{\text{cho}}, a_{\text{rej}}) \sim \mathcal{D}_{\text{DPO}}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(a_{\text{cho}}|q_i)}{\pi_{ref}(a_{\text{cho}}|q_i)} - \beta \log \frac{\pi_\theta(a_{\text{rej}}|q_i)}{\pi_{ref}(a_{\text{rej}}|q_i)} \right) \right]$$

$$\mathcal{L}_{\text{CE}}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(q_i, a_{\text{cho}}) \sim \mathcal{D}} \left[\log \left(\pi_\theta(a_{\text{cho}}|q_i) \right) \right]$$

$$\mathcal{L}_{\text{merge}} = \lambda \cdot \mathcal{L}_{\text{DPO}} + \mathcal{L}_{\text{CE}}$$

In this context, λ represents the coefficient of the cross-entropy loss for the chosen answer in the total loss. Commonly, we experiment with values in $\{0.5, 1, 1.5\}$. Another critical coefficient is β, which measures the penalty intensity of DPO for incorrect answers. Owing to the addition of a regularization term, the value of this coefficient is higher than that of the standard DPO, with our testing range for this value being $\{0.5, 1, 2\}$. Besides these, the overall learning rate is set at 1e-6. The experimental section will report the optimal results under these coefficient settings. We train 500 steps with a batch size of 64 in this stage.

Table 4: The composition of the MathUserEval data set. We divided the test set into three categories: Elementary and Advanced Mathematics. For calculating the total score, we used the macro-average score. Dia. refer to dialogues.

Category	Sub-Category	Size	Source
Elementary	Calculate	75	Dia.
	Algebra	113	
	Geometry	81	
	Trigonometry	73	
Advanced	Discrete Math	45	Dia.&Exams
	Probability	46	
	Linear Algebra	58	
	Calculus	54	

5 MathUserEval: Benchmarking LLMs' Math Reasoning in Application

MathUserEval is a test set designed for real-use scenarios, addressing user concerns and more challenging mathematical problems. Some data originates from university examination questions, while others come from simulated dialogues. In the latter, annotators posed math-related questions using large models based on their daily experiences and observations.

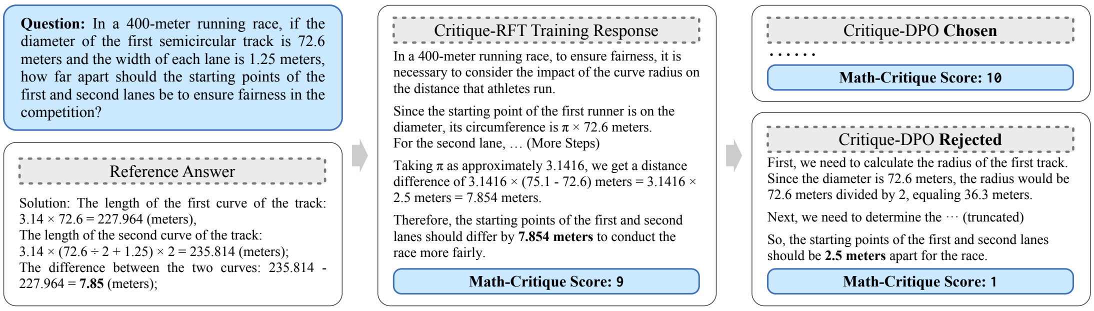
Based on the distribution of the collected data, we divided the test set into two main categories, Elementary and Advanced, and eight sub-categories. Given that Calculate Applications are less challenging and closely aligned with the scope of previous public datasets, we selected fewer questions from this category. The quantity of questions in each of these categories is as shown in Table 4. All questions are posed in an open-ended format. Possible answers include a single number, multiple numbers, or mathematical expressions.

We offer two evaluation methods: GPT-4-1106-Preview (OpenAI, 2023; Liu et al., 2023a; Zheng et al., 2023) evaluation and Math-Critique evaluation. The former adopts the alignbench (Liu et al., 2023a) evaluation method for a more accurate, fair, and accessible approach; the latter uses the Math-Critique method described earlier.

6 Experiment

6.1 Data Collection

The primary sources of our data collection include public datasets and publicly available middle school and university examination questions. We selected English data prompts from the GSM8k and



(1) Given question and reference answer (2) An example response adopted by RFT stage (3) An example pair adopted by DPO stage

Figure 2: Training data examples. The data we generate is divided into two categories, originating from the questions and references within existing datasets. We have constructed separate data for RFT and pairwise DPO training.

MATH training sets, using the original dataset responses as standard answers. We used the provided answer formats as the common answers for publicly available middle school and university exam questions without further processing. Details of our training data are provided in Appendix F.

6.2 Evaluation Setting

Evaluation Datasets. In our research, we primarily tested the MathUserEval dataset, derived from simulated dialogue records and actual exam papers, offering diverse question styles and real-world relevance. Additionally, we tested academic datasets: GSM8k (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b) for English, and Ape210k (Zhao et al., 2020) and Cmath (Wei et al., 2023b) for Chinese. We also used the Hungarian National Exam (Paster, 2023) as an Out-Of-Distribution test set, and the Chinese language component of AlignBench (Liu et al., 2023a) and full MT-Bench (Zheng et al., 2023) to evaluate general linguistic capabilities.

Base Model. Since we couldn't determine whether the open-source models had undergone instruction fine-tuning specifically for the math domain, we chose ChatGLM3-32B-SFT-2312 as our base model for training. This model has been thoroughly pre-trained but only partially fine-tuned with instructions. Additionally, we carefully removed all instruction data related to solving math problems.

Baselines. Since most of our work is conducted in Chinese, we selected three categories of baselines: open-source mathematics models, open-source Chinese models, and leading proprietary models. For the open-source mathematics models, we chose SkyMath (Yang et al., 2023b), MetaMath (Yu et al., 2023), and Internlm2-Math (Team, 2023). To effectively compare with the best Chinese models, we selected Qwen-Chat (Bai et al., 2023a), Yi-Chat (Yi, 2023), DeepSeek-Chat (DeepSeek-AI et al., 2024), and InternLM2 (Team, 2023). We also report the results for GPT-4-1106-Preview, GPT-4-0613, GPT-3.5-Turbo (OpenAI, 2023), and Claude-2 (Anthropic, 2023).

Metrics. For all datasets, we used the results of greedy inference performed once. For academic datasets, we report self-reported results of corresponding models and the highest zero-shot/few-shot results from the OpenCompass and MathUserEval websites. For the math subset of AlignBench (Liu et al., 2023a) and our MathUserEval test set, we report scoring results from GPT-4-Turbo and Math-Critique. More details on evaluation settings are in Appendix C.

6.3 Main Results

Table 5 shows that our model scored 4.23 on MathUserEval, 89.4 on Ape210k, and 40.6 on MATH, surpassing all published models and achieving near-top performances on Cmath and GSM8k. Additionally, it scored 73 on the Hungarian Test, the highest among known parameter models.

Using ChatGLM3-32B-SFT-2312 as our baseline, the RFT phase significantly improved performance across all math datasets, while the DPO phase enhanced performance on open-ended math problems like MathUserEval, the Hungarian Exam, and AlignBench. Despite minimal improvement on MT-Bench, parity was maintained, preserving English capabilities given the predominantly Chinese training data.

Compared to proprietary models like OpenAI's GPT series, GLM-4 demonstrates competitive or superior performance, surpassing GPT-4-1106-Preview in the Ape210k and AlignBench benchmarks, indicating strengths in mathematical reason-

Table 5: Main Result. All results reported are the highest achieved in zero-shot or few-shot settings and are based on greedy decoding. The best models are marked in **bold** and the underline signifies the second best model.

Models	#params	Chinese					English			General	
		MathUserEval			Ape210k	Cmath	GSM8k	MATH	Hunga-rian	AlignBench	MT-Bench
		Overall	Elementary	Advanced						Language	
GPT-4-1106-Preview (OpenAI, 2023)	N/A	**5.73**	**5.07**	**6.81**	84.2	**89.3**	**93.6**	**53.6**	**92**	8.29	**9.32**
GPT-4-0613 (OpenAI, 2023)	N/A	4.14	3.34	5.33	83.6	86.5	91.4	45.8	68	7.59	9.18
GPT-3.5-Turbo-0613 (OpenAI, 2023)	N/A	3.42	3.04	4.07	70.4	76.8	78.2	28.0	41	6.82	8.36
Claude-2 (Anthropic, 2023)	N/A	3.29	2.63	4.35	72.8	80.5	88.0	-	55	6.78	8.06
GLM-4	N/A	5.11	4.86	5.43	**93.5**	89.0	91.8	49.0	75	**8.38**	8.62
Skywork-13B-Math (Yang et al., 2023b)	13B	2.66	2.75	2.54	74.4	77.3	72.3	17.0	39	5.58	4.12
InternLM2-Chat (Team, 2023)	20B	3.25	3.00	3.68	72.0	80.7	79.6	34.8	48	7.68	8.21
Math-InternLM2 (Team, 2023)	20B	3.17	3.08	3.37	75.2	78.5	82.6	37.7	66	6.53	6.09
Yi-Chat (Yi, 2023)	34B	2.64	2.49	2.87	65.1	77.7	76.0	15.9	39	6.18	6.54
DeepSeek-Chat (DeepSeek-AI et al., 2024)	67B	3.24	2.76	3.84	76.7	80.3	**84.1**	32.6	58	7.11	**8.35**
MetaMath (EN) (Yu et al., 2023)	70B	-	-	-	-	-	82.3	26.0	35	-	4.28
Qwen-Chat (Bai et al., 2023a)	72B	3.87	3.99	3.67	77.1	**88.1**	76.4	31.8	52	7.29	6.43
ChatGLM3-32B-SFT-2312*	32B	3.25	3.03	3.60	78.0	79.8	75.8	29.0	39	7.37	8.05
+ RFT	32B	4.01	3.86	4.26	87.0	85.3	82.4	39.5	58	7.42	8.03
+ RFT, DPO	32B	**4.23**	**4.01**	**4.59**	**89.4**	85.6	82.6	**40.6**	**73**	**7.80**	8.08
		+0.98	+0.98	+0.99	+11.4	+5.8	+6.8	+11.6	+34.0	+0.43	+0.03

Table 6: Ablation Study for 32B model. All results are fine-tuned from our 32B base model. We selected Metamath training set as baselines that we consider comparatively strong. MATHUSEREVAL is scored with Math-Critique model.

Method	Chinese		English	
	MATHUSEREVAL	Ape210k*	GSM8k	MATH*
Metamath (Yu et al., 2023)	2.80	75.8	77.9	35.6
ChatGLM3-32B-SFT + RFT	3.74	87.0	82.4	39.5
- Real scenarios & Academic	3.29	85.9	74.8	27.6
- Real scenarios	3.29	74.6	77.4	36.0
- Academic	3.72	75.8	81.0	36.2
ChatGLM3-32B-SFT + RFT & DPO	4.37	89.4	82.6	41.0
- Real Scenarios & Academic	4.14	87.8	81.5	37.8

* Ablated experiments are conducted on 500-sample test subsets.

ing and cross-linguistic generalization.

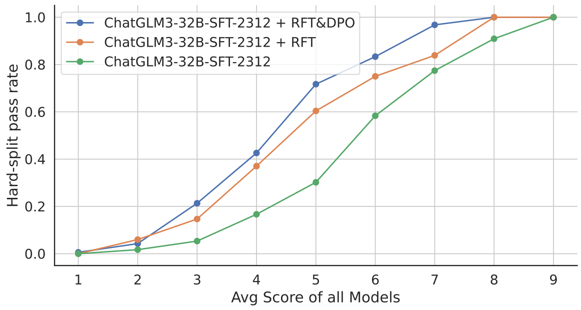


Figure 3: The Relationship between Different Boosting Methods and Problem Difficulty. The horizontal axis displays the average score of MATHUSEREVAL across 24 models (scored by GPT-4-1106-Preview), which we regard as a representation of problem difficulty. The vertical axis represents the hard-split scores of the models on these questions.

6.4 Analysis

Ablation of data composition. Table 6 presents the results of ablation experiments using the Metamath (Yu et al., 2023) training set as a baseline. After applying Critique-RFT, we found that using only academic datasets resulted in poorer performance on real-life scenario-based MATHUSEREVAL and academic test sets compared to integrating real-life scenario data. Additionally, introducing English data significantly improved performance on English datasets without negatively impacting Chinese capabilities.

During the Critique-DPO phase, the ablation experiments showed that adding math-specific DPO data significantly enhances mathematical capabilities compared to using general DPO data. We did not test the impact of Real scenarios and Academic data separately, as questions that the model could solve were removed in previous stages, leaving insufficient data for a complete training session.

Relationship between Different Boosting Methods and Problem Difficulty. Figure 3 illustrates the relationship between the average accuracy of each question in MATHUSEREVAL across all 24 models tested (including some intermediate models) and the hard-split scores of the four GLM series models. The average accuracy is considered indicative of the question's difficulty level. The RFT step improves performance across almost all difficulty levels, with the most significant gains for questions averaging scores between 4 and 6. The DPO step mainly enhances performance on questions with average scores between 5 and 7.

Impact on general capabilities. To develop a general model with strong mathematical capabilities, we evaluated our results using Alignbench (Liu et al., 2023a), showing our model surpasses similar

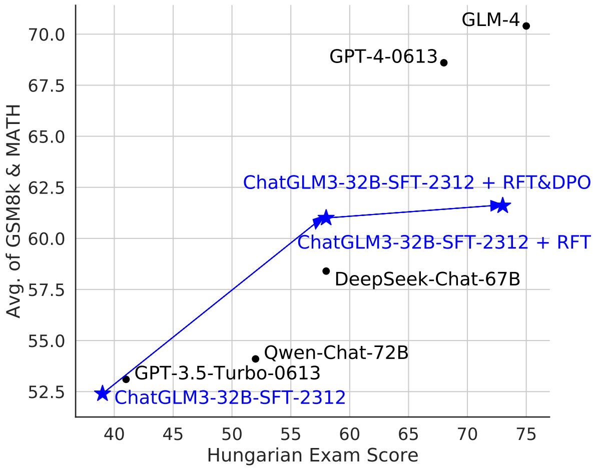


Figure 4: Results of Hungarian Exam and Average Scores of GSM8k and MATH.

Table 7: Evaluation for Math-Critique Model. We report "Acc" as the accuracy of the model in determining whether an answer is correct, as well as the Pearson, Spearman, and Kendall correlation coefficients for Math-Critique in comparison with human annotations in a four-category classification.

Model	Acc.	Pearson	Spearman	Kendall
GPT-3.5-Turbo	62.1	31.8	33.5	30.1
GPT-4-0613	90.2	80.5	78.1	71.0
Math-Critique-32B	90.5	80.4	77.1	70.2

baselines in Chinese language capabilities and excels compared to other Chinese mathematical and general models (Table 5). Using MT-Bench (Zheng et al., 2023) for English general capabilities, we found that despite over 90% of our training data being in Chinese, our model's English performance remained largely unaffected.

Effectiveness of Math-Critique. In evaluating Math-Critique's effectiveness, we annotated an 800-question test set into four categories and validated it against Chinese high school exams and MATHUSEREVAL. Empirical experiments showed that Math-Critique-32B outperformed GPT-3.5-Turbo in judgment accuracy and correlation with human annotations, comparable to GPT-4-0613, as shown in Table 7. More details are shown in Appendix D.4.

Comparsion with tool-using models. Tool-using aids in solving difficult math problems, but fine-tuning on tool-using data harms LLM's general abilities. As shown in Table 8, we tested similar-sized models using tool or code calls, including Mammoth-34B (Yue et al., 2023), Tora-34B (Gou et al., 2023), and Openmath-34B (Toshniwal et al., 2024), on GSM8k, MATH, and mt-bench datasets. Our model significantly surpasses these in terms of language capabilities. Although all use CodeLLaMA-34B as the base model, CodeLLaMA-34B-instruct's mt-bench score is 6.47, suggesting their lower scores might result from poor timing in external tool usage, impacting their general domain response abilities.

Table 8: Performance comparison with tool-using models. Note that Mammoth-34b, Tora-34b, and Openmath-34b are trained based on CodeLLaMA-34b.

Model	GSM8k	MATH	MT-Bench
Mammoth-34B (Yue et al., 2023)	72.7	43.6	5.45
Tora-34B (Gou et al., 2023)	80.7	51.0	4.27
Openmath-34B (Toshniwal et al., 2024)	80.7	48.3	2.32
CodeLLaMA-34B (Rozière et al., 2024)	53.3	23.9	6.47
ChatGLM3-32B-SFT-2312 + RFT & DPO	82.6	40.6	8.08

Out-Of-Distribution Test. Following the Grok-1 approach, we evaluated our model's mathematical capabilities on the Hungarian national final exam (Paster, 2023), an OOD dataset with 33 questions. As shown in Figure 4, human expert evaluation revealed scores of 57 for the 32B RFT model and 73 for the DPO model. Notably, correct answers in Chinese were scored appropriately, considering the model's primary language.

Relationship between RFT and DPO phrase. While the gains from the DPO phase may seem smaller than the RFT phase, two main factors explain this. First, RFT is a simplified version of DPO, so extensive learning during RFT reduces the visible improvements in the DPO phase. Second, by enhancing generalization, DPO significantly boosts performance on Out Of Domain (OOD) test sets, like MathUserEval and Hungarian-exam. Additionally, DPO consolidates the model's understanding of previously learned problems, ensuring more accurate responses within its capabilities.

7 Conclusion

In this paper, we present Math-Critique, a method for evaluating mathematical problem correctness, and Self-Critique, aimed at enhancing language models' mathematical skills without supervision. Our experiments in English and Chinese show a 32B parameter model achieving state-of-the-art results among open-source models, surpassing proprietary models like GPT-4-0613 on the MATHUSEREVAL test set. This approach was key in developing GLM-4, boosting performance on datasets such as MATHUSEREVAL, GSM8k, and the Hungarian test.

Limitations

Graphic thinking and drawing abilities. Our purely linguistic model struggles with tasks requiring drawing or understanding images. For instance, it correctly lists, but cannot accurately draw, a topology of numbers requiring divisors in the Hungary Test. Integrating multimodal input and output components could address these issues, which is a direction we plan to explore.

Precision calculation capability. The model may deviate up to 5% for problems needing multiplication, division, or exponentiation of numbers with three or more decimal places, similar to GPT-4 without an integrated code interpreter. This fundamental limitation might not be resolved by merely increasing the model size. Utilizing external computation tools or code interpreters can solve this problem, though our focus here remains on pure language models. We aim to tackle these precision issues in future work.

Acknowledgements

This work is supported by the Natural Science Foundation of China(NSFC) 62276148 Tsinghua University (Department of Computer Science and Technology) -Siemens Ltd., China Joint Research Center for Industrial Intelligence and Internet of Things (JCIIOT) and New Cornerstone Science Foundation through the XPLORER PRIZE.

References

Anthropic. 2023. Introducing claude.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. 2021. A general language assistant as a laboratory for alignment.

Jinze Bai, Shuai Bai, et al. 2023a. Qwen technical report.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023b. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

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

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439.

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

Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2023. Black-box prompt optimization: Aligning large language models without model training.

Ethan Chern, Haoyang Zou, Xuefeng Li, Jiewen Hu, Kehua Feng, Junlong Li, and Pengfei Liu. 2023. Generative ai for math: Abel. `https://github.com/GAIR-NLP/abel`.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

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

DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. Deepseek llm: Scaling open-source language models with longtermism.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Yao Fu, Litu Ou, Mingyu Chen, Yuhao Wan, Hao Peng, and Tushar Khot. 2023. Chain-of-thought hub: A continuous effort to measure large language models' reasoning performance. *arXiv preprint arXiv:2305.17306*.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Tora: A tool-integrated reasoning agent for mathematical problem solving.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Zhenyu Hou, Yiin Niu, Zhengxiao Du, Xiaohan Zhang, Xiao Liu, Aohan Zeng, Qinkai Zheng, Minlie Huang, Hongning Wang, Jie Tang, and Yuxiao Dong. 2024. Chatglm-rlhf: Practices of aligning large language models with human feedback.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Pei Ke, Bosi Wen, Zhuoer Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Shengyuan Wang, Aohan Zeng, Yuxiao Dong, Hongning Wang, et al. 2023. Critiquellm: Scaling llm-as-critic for effective and explainable evaluation of large language model generation. *arXiv preprint arXiv:2311.18702*.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. Mawps: A math word problem repository. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, pages 1152–1157.

Vivek Kumar, Rishabh Maheshwary, and Vikram Pudi. 2021. Adversarial examples for evaluating math word problem solvers. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2705–2712.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback.

Yudong Li, Yuqing Zhang, Zhe Zhao, Linlin Shen, Weijie Liu, Weiquan Mao, and Hui Zhang. 2022. CSL: A large-scale Chinese scientific literature dataset. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3917–3923. International Committee on Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Zhuoer Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, et al. 2023a. Alignbench: Benchmarking chinese alignment of large language models. *arXiv preprint arXiv:2311.18743*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2023b. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. Numglue: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3505–3523.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Keiran Paster. 2023. Testing language models on a held-out high school national finals exam. `https://huggingface.co/datasets/keirp/hungarian_national_hs_finals_exam`.

Stanislas Polu and Ilya Sutskever. 2020. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*.

Jinghui Qin, Xiaodan Liang, Yining Hong, Jianheng Tang, and Liang Lin. 2021. Neural-symbolic solver for math word problems with auxiliary tasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5870–5881, Online. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2024. Code llama: Open foundation models for code.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities. `https://github.com/InternLM/InternLM`.

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. 2024. Openmathinstruct-1: A 1.8 million math instruction tuning dataset.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Quan Tu, Shilong Fan, Zihang Tian, and Rui Yan. 2024. Charactereval: A chinese benchmark for role-playing conversational agent evaluation.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. 2023. Math-shepherd: Verify and reinforce llms step-by-step without human annotations.

Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854, Copenhagen, Denmark. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023a. Chain-of-thought prompting elicits reasoning in large language models.

Tianwen Wei, Jian Luan, Wei Liu, Shuang Dong, and Bin Wang. 2023b. Cmath: Can your language model pass chinese elementary school math test?

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023a. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Liu Yang, Haihua Yang, Wenjun Cheng, Lei Lin, Chenxia Li, Yifu Chen, Lunan Liu, Jianfei Pan, Tianwen Wei, Biye Li, Liang Zhao, Lijie Wang, Bo Zhu, Guoliang Li, Xuejie Wu, Xilin Luo, and Rui Hu. 2023b. Skymath: Technical report.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Yi. 2023. A series of large language models trained from scratch by developers at 01-ai. `https://github.com/01-ai/Yi`.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Beichen Zhang, Kun Zhou, Xilin Wei, Wayne Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. 2023a. Evaluating and improving tool-augmented computation-intensive math reasoning. *arXiv preprint arXiv:2306.02408*.

Dan Zhang, Ziniu Hu, Sining Zhoubian, Zhengxiao Du, Kaiyu Yang, Zihan Wang, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Sciglm: Training scientific language models with self-reflective instruction annotation and tuning.

Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2023b. Evaluating the performance of large language models on gaokao benchmark.

Wei Zhao, Mingyue Shang, Yang Liu, Liang Wang, and Jingming Liu. 2020. Ape210k: A large-scale and template-rich dataset of math word problems.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models.

Jinfeng Zhou, Zhuang Chen, Dazhen Wan, Bosi Wen, Yi Song, Jifan Yu, Yongkang Huang, Libiao Peng, Jiaming Yang, Xiyao Xiao, et al. 2023a. Characterglm: Customizing chinese conversational ai characters with large language models. *arXiv preprint arXiv:2311.16832*.

Zihao Zhou, Qiufeng Wang, Mingyu Jin, Jie Yao, Jianan Ye, Wei Liu, Wei Wang, Xiaowei Huang, and Kaizhu Huang. 2023b. Mathattack: Attacking large language models towards math solving ability. *arXiv preprint arXiv:2309.01686*.

A Case Study

A.1 Case Study of Math-Critique

We have provided several examples of scoring by Math-Critique. In the examples from Table 9 and Table 10, we observed that the model provided correct answers. However, the answer formats differed

Table 9: Math-Critique Example (1)

Question	1/x+x三次方的原函数是多少。
Question (Translate)	What is the original function of 1/x+x cubed.
Reference	首先，我们可以将 $1/x+x^3$ 重新写成 $x^{-1}+x^3$，然后利用幂函数和多项式函数的求导原理求出它的原函数。 具体来说，我们得到: $$\int(x^{-1}+x^3)\mathrm{d}x=\int x^{-1}\mathrm{d}x+\int x^3\mathrm{d}x=\ln\lvert x\rvert+\frac{1}{4}x^4+C,$$ 其中 C 是不定积分常数。 因此，$1/x+x^3$ 的原函数是 $\ln\lvert x\rvert+\frac{1}{4}x^4+C$。
Reference (Translate)	First, we can rewrite $1/x+x^3$ as $x^{-1}+x^3$, and then use the derivation principles of power functions and polynomial functions to find its original function. Specifically, we get: $$\int(x^{-1}+x^3)\mathrm{d}x=\int x^{-1}\mathrm{d}x+\int x^3\mathrm{d}x=\ln\lvert x\rvert+\frac{1}{4}x^4+C,$$ where C is the indefinite integral constant. Therefore, the original function of $1/x+x^3$ is $\ln\lvert x\rvert+\frac{1}{4}x^4+C$.
Model Answer by GLM4	要求函数 $f(x)=\frac{1}{x}+x^3$ 的原函数（不定积分），我们需要分别对这两项进行积分。 对于 $\frac{1}{x}$，其原函数是 $\ln\lvert x\rvert$（自然对数）。对于 x^3，我们可以利用幂函数的积分规则，即对于 x^n（其中 $neq-1$），其原函数是 $\frac{x^{n+1}}{n+1}$。因此，x^3 的原函数是 $\frac{x^4}{4}$。将这两项的原函数相加，我们得到 $f(x)$ 的原函数: $$\int\left(\frac{1}{x}+x^3\right)dx=\ln\lvert x\rvert+\frac{x^4}{4}+C$$ 其中 C 是积分常数。这个结果表示在定义域内（除了 $x=0$ 的情况，因为 $\frac{1}{x}$ 在 $x=0$ 处未定义），$f(x)$ 的一个原函数。由于 $\ln\lvert x\rvert$ 在 $x=0$ 处未定义，所以这个原函数在 $x=0$ 处也是不连续的。
Model Answer (Translate)	To find the primitive function (indefinite integral) of the function $f(x)=\frac{1}{x}+x^3$, we need to integrate these two terms separately. For $\frac{1}{x}$, the original function is $\ln\lvert x\rvert$ (natural logarithm). For x^3, we can use the integration rule of the power function, that is, for x^n (where $neq-1$), the original function is$\frac{x^{n+1}}{n+1}$. Therefore, the original function of x^3 is $\frac{x^4}{4}$. Adding the original functions of these two terms, we get the original function of $f(x)$: $$\int\left(\frac{1}{x}+x^3\right)dx=\ln\lvert x\rvert+\frac{x^4}{4}+C$$ where C is the integration constant. This result means that within the domain (except for the case of $x=0$, since $\frac{1}{x}$ is undefined at $x=0$), $f(x) A primitive function of$. Since $\ln\lvert x\rvert$ is undefined at $x=0$, the original function is also discontinuous at $x=0$.
Critique Result	AI助手的回答非常准确和详细，它正确地解释了如何求解给定函数的原函数。AI助手首先分别对函数的两部分进行积分，然后将它们的结果相加，得到了原函数。。因此，我会给它一个高分。评分：[[9]]
Critique Result (Translate)	The AI assistant's answer is very accurate and detailed, and it correctly explains how to solve the original function of a given function. The AI assistant first integrates the two parts of the function separately and then adds their results to obtain the original function. Therefore, I would give it a high score. Rating:[[9]]

Table 10: Math-Critique Example (2)

Question	三阶矩阵A有三个特征值0,3,5，对应的特征向量分别为u,v,w，则线性方程组Ax = v + w的通解为?
Question (Translate)	The third-order matrix A has three eigenvalues 0, 3, and 5, and the corresponding eigenvectors are u, v, and w respectively. What is the general solution to the linear equation system Ax = v + w?
Reference	给定矩阵A有特征值0,3,5，对应的特征向量为u,v,w。线性方程组Ax = v + w，我们可以写作Au = 0 * u，Av = 3 * v，和Aw = 5 * w。所以A(1/3*v)=v,A(1/5*w)=w，左右两边相加得A(1/3*v+1/5*w)= v + w。又因为Ax = v + w，所以1/3 * v + 1/5 * w是Ax = v + w的一个特解，因为Au = 0，所以u是Au = 0的基础解系，因此方程Ax = v + w的通解是x =ku+ 1/3 * v + 1/5 * w。
Reference (Translate)	A given matrix A has eigenvalues 0, 3, and 5, and the corresponding eigenvectors are u, v, and w. For the linear system of equations Ax = v + w, we can write Au = 0 * u, Av = 3 * v, and Aw = 5 * w. So A(1/3*v)=v, A(1/5*w)=w, adding the left and right sides gives A(1/3*v+1/5*w)= v + w. And because Ax = v + w, 1/3 * v + 1/5 * w is a special solution of Ax = v + w. Because Au = 0, so u is the basic solution system of Au = 0, so the equation Ax The general solution of = v + w is x =ku+ 1/3 * v + 1/5 * w.
Model Answer by ChatGLM3-32B-Math	给定三阶矩阵 A 的三个特征值分别为 $0, 3, 5$，对应的特征向量分别为 u, v, w。要找线性方程组 $Ax = v + w$ 的通解，我们可以利用特征值和特征向量的性质。 首先，我们知道 $Av = 3v$ 和 $Aw = 5w$。因此，v 和 w 分别是特征值 3 和 5 对应的特征向量。 线性方程组 $Ax = v+w$ 可以重写为 $Ax-v-w = 0$。我们可以将 x 写成 $x = pu + qv + rw$，其中 p, q, r 是待定系数。 将 x 代入方程 $Ax - v - w = 0$，我们得到: $$A(pu + qv + rw) - v - w = 0$$ ... 由于 v 和 w 是线性无关的特征向量（因为它们对应不同的特征值），上式成立的唯一可能是 $3q-1 = 0$ 和 $5r - 1 = 0$。 解这两个方程，我们得到 $q = \frac{1}{3}$ 和 $r = \frac{1}{5}$。 因此，方程 $Ax = v + w$ 的通解为: $$x = pu + \frac{1}{3}v + \frac{1}{5}w$$ 其中 p 是任意实数。 所以，通解的形式是: $$x = p\mathbf{u} + \frac{1}{3}\mathbf{v} + \frac{1}{5}\mathbf{w}$$ 这里的 $\mathbf{u}$ 是特征值 0 对应的特征向量，$\mathbf{v}$ 和 $\mathbf{w}$ 分别是特征值 3 和 5 对应的特征向量。p 是任意实数，可以取任何值。
Model Answer (Translate)	Given the three eigenvalues of the third-order matrix A are $0, 3, 5$, the corresponding eigenvectors are u, v, w. To find the general solution to a system of linear equations $Ax = v + w$, we can exploit the properties of eigenvalues and eigenvectors. First of all, we know $Av = 3v$ and $Aw = 5w$. Therefore, v and w are the eigenvectors corresponding to the eigenvalues 3 and 5 respectively. The system of linear equations $Ax = v + w$ can be rewritten as $Ax - v - w = 0$. We can write x as $x = pu + qv + rw$, where p, q, r are the undetermined coefficients. Substituting x into the equation$Ax - v - w = 0$, we get: $$A(pu + qv + rw) - v - w = 0$$ ... Therefore, the general solution of equation $Ax = v + w$ is: $$x = pu + \frac{1}{3}v + \frac{1}{5}w$$ Where p is any real number. So, the general solution is in the form: $$x = p\mathbf{u} + \frac{1}{3}\mathbf{v} + \frac{1}{5}\mathbf{w}$$ Here$\mathbf{u}$ is the eigenvector corresponding to the eigenvalue 0, $\mathbf{v}$ and $\mathbf{w}$ are the eigenvectors corresponding to the eigenvalues 3 and 5 respectively. p is any real number and can take on any value.
Critique Result	AI助手的回答非常详细，它首先解释了如何利用特征值和特征向量的性质来解决这个问题，然后逐步进行了详细的步骤，包括将方程重写，将x写成待定系数的形式，然后将这些系数代入方程，最后解出这些系数。这个答案不仅准确，而且深入，提供了详细的解释和步骤，使得读者能够理解如何解决这个问题。与参考答案相比，AI助手的答案更加详细和深入，因此我认为AI助手的回答质量非常高。评分：[[9]]。
Critique Result (Translate)	The AI assistant's answer is very detailed. It first explains how to use the properties of eigenvalues and eigenvectors to solve this problem, and then goes through the detailed steps, including rewriting the equation, writing x in the form of undetermined coefficients, and then converting these The coefficients are substituted into the equation and finally the coefficients are solved. This answer is not only accurate but also in-depth, providing detailed explanations and steps so the reader can understand how to solve the problem. Compared with the reference answers, the AI assistant's answers are more detailed and in-depth, so I think the quality of the AI assistant's answers is very high. Rating: [[9]].

Table 11: Math-Critique Example (3)

Question	二元一次方程4x-3y=12,当x=0,1,2,3时,y=
Question (Translate)	Quadratic equation 4x-3y=12, when x=0,1,2,3, y=
Reference	将二元一次方程4x-3y=12改写为y的表达式： 4x-3y=12 -3y = -4x + 12 y = (4/3)x - 4 当x=0时，y=(-4)； 当x=1时，y=(4/3)-4=-8/3； 当x=2时，y=(8/3)-4=-4/3； 当x=3时，y=(4)-4=0。 因此，当x分别为0、1、2、3时，y的值分别为-4、-8/3、-4/3和0。
Reference (Translate)	Rewrite the linear equation 4x-3y=12 into the expression of y: 4x-3y=12 -3y = -4x + 12 y = (4/3)x - 4 When x= When 0, y=(-4); When x=1, y=(4/3)-4=-8/3; When x=2, y=(8/3)-4 =-4/3; When x=3, y=(4)-4=0. Therefore, when x is 0, 1, 2, and 3 respectively, the values of y are -4, -8/3, -4/3, and 0 respectively.
Model Answer by Claude-2	* 给定二元一次方程:4x - 3y = 12 * 将x替换为给定的数值,可以得到y的值: * x = 0 代入方程,得到: -3y = 12, y = -4 * x = 1 代入方程,得到: 4 - 3y = 12, -3y = 8, y = -2 * x = 2 代入方程,得到: 8 - 3y = 12, -3y = 4, y = 0 * x = 3 代入方程,得到:12 - 3y = 12, -3y = 0, y = 2 * 综上,当x=0,1,2,3时,y依次等于-4,-2,0,2。 所以应该填写:-4,-2,0,2。
Model Answer (Translate)	* Given a linear equation of two variables: 4x - 3y = 12 * Replace x with the given value, you can get the value of y: * x = 0 Substitute into the equation, you get: -3y = 12, y = - 4 * x = 1 Substituting into the equation, we get: 4 - 3y = 12, -3y = 8, y = -2 * x = 2 Substituting into the equation, we get: 8 - 3y = 12, -3y = 4, y = 0 * x = 3 Substituting into the equation, we get: 12 - 3y = 12, -3y = 0, y = 2 * In summary, when x=0,1,2,3, y is equal to -4 in turn ,-2,0,2. So you should fill in:-4,-2,0,2.
Critique Result	AI助手的回答在计算过程中出现了错误。当x=1时，AI助手错误地计算出y=-2，而实际上应该是y=-8/3。同样，当x=3时，AI助手错误地计算出y=2，而实际上应该是y=0。因此，AI助手的回答在准确性和深度上都存在问题。然而，AI助手的回答在结构和详细程度上是正确的，它详细地解释了如何计算y的值，并给出了每个x值对应的y值。总的来说，AI助手的回答虽然有一些错误，但是仍然提供了一些有用的信息。因此，我会给AI助手的回答打分为3。 评分：[[3]]
Critique Result (Translate)	There was an error in the calculation process of the AI assistant's answer. When x=1, the AI assistant incorrectly calculates y=-2, when it should actually be y=-8/3. Likewise, when x=3, the AI assistant incorrectly calculates y=2, when it should actually be y=0. Therefore, the AI assistant's answers have problems with accuracy and depth. However, the AI assistant's answer is correct in structure and level of detail. It explains in detail how to calculate the value of y and gives the corresponding y value for each x value. Overall, although the AI assistant's answers contained some errors, they still provided some useful information. Therefore, I would rate the AI assistant's answer a 3. Rating: [[3]]

from the standard answers, specifically regarding fraction expression and the selection of unknown variables. These differences are equivalent, yet traditional evaluation methods struggle to judge them accurately. However, Math-Critique correctly scored these two examples and provided reasonable evaluations.

In the example from Table 11, the model made a mistake in the calculation process. Math-Critique accurately pinpointed the error location, and since the model correctly solved a part of the problem, Math-Critique awarded a score of 3 points.

A.2 Case Study of Mathematical Models

Here are a few comparisons between ChatGLM3-32B-Math(ChatGLM3-32B-SFT-2312 + RFT&DPO) and other models. In the example from Table 12, the problem is a math question of Chinese junior high school difficulty. During the solution process by GPT-4-0613, an error occurred in solving the equation. ChatGLM3-32B-SFT-2312 did not correctly understand the question. ChatGLM3-32B-Math correctly listed the equation and accurately solved it using the factorization method.

In the example from Table 13, both GPT-4-0613 and ChatGLM3-32B-Math provided the correct answers, but the difference lies in that ChatGLM3-32B-Math offered a very detailed derivation process. We believe that detailed derivation aids in understanding for users and helps prevent errors that may occur during the model's step-skipping.

In the example from Table 14, originating from the Hungry Test, ChatGLM3-32B-Math correctly conducted the analysis and provided the solution. In contrast, Qwen-Max, despite being accurate in most processes, made a simplification error in the expressions for S6 and S7, leading to an incorrect result despite precise calculations.

A.3 Case Study of Errors

In the first error example shown in Table 15, ChatGLM3-32B-Math correctly listed the matrix equations but made a mistake in transferring terms during the complex calculation process. This led to an incorrect solution despite the overall process being correct.

In the second example shown in Table 16, the question includes an image. This question originates from MATH, and although the image is provided in Asymptote code, the language model still struggles to understand the meaning of the image.

```
Math-Critique Prompt Template
[Instruction]
请你扮演一名评审，对人工智能助手对数学问题生成的回答进行评分。你的评价需要优先考虑最终答案的正确性，其次关注解题步骤每一步的正确性，并严格按照下面的评分标准进行评分：
1)分数：1~2， 标准：最终答案完全错误，并且解答过程中错误出现很早，即解答刚开始便出现明显错误。
2)分数：3~5， 标准：最终答案错误，且解题过程前半部分正确，但后半部分中出现错误。
3)分数：6~8， 标准：最终答案正确，但中间步骤出现部分错误。
4)分数：9~10， 标准：最终答案完全正确，且解题步骤完全正确。
我会提供一段参考答案和人工智能助手的答案，请你以参考答案为基础对人工智能助手的答案尽可能客观地评价。请你输出 1) 首先提供一段简短的解释，用来评价人工智能助手回答的质量，如有最终答案错误或者步骤错误，请指出并简单解释；2) 然后给出评估分数，必须严格按照以下格式进行评分：\"[[rating]]\"，例如：\"评分:[[4]]\".
[Question]
设X服从泊松分布P(λ)，且E[(X-1)(X-2)]=1，则λ等于?
[The Start of Reference Answer]
首先我们需要知道泊松分布的期望是λ，方差是λ。由题意，E[(X-1)(X-2)]=1，所以 E[(X-1)(X-2)]=E[X^2]-3E[X]+2=1
DX=EX^2-EX^2
EX^2=λ+λ^2
解这个方程我们可以得到λ=1
[The End of Reference Answer]
[The Start of Assistant's Answer]
设随机变量X服从泊松分布P(λ)，我们知道泊松分布的期望值E[X]等于λ，方差Var[X]也等于λ。
根据题目条件，我们有:
E[(X-1)(X-2)] = 1
因为X服从泊松分布，我们可以利用二项式定理展开(X-1)(X-2)，然后将其与期望值相乘以找到λ。
(X-1)(X-2) = X^2 - 3X + 2
展开后得到:
E[X^2 - 3X + 2] = 1
我们知道E[X^2] = Var[X] + (E[X])^2，且E[X] = λ。因此，我们可以将E[X^2]替换为λ^2 + λ，并且知道E[X] = λ。
所以，我们有:
λ^2 + λ - 3λ + 2 = 1
解这个方程得到 λ = 1。
所以，λ的值为1。
[The End of Assistant's Answer]
```

Figure 5: Prompt Template with an example for Math-Critique Model.

This demonstrates a deficiency in our model's ability to process images. In fact, within the MATH test set, for questions that include images, our model's accuracy rate is only 23%, significantly lower than the 40% accuracy rate across the entire dataset.

In the third example shown in Table 17, we can observe certain deficiencies in the computational accuracy of the language model. In the first column of the table, the model correctly calculates the multiplication of integers and a single decimal. Still, in the second column, when calculating 424*1.06, the model incurred an error of approximately 0.2%. This error led to subsequent calculations being incorrect due to the accumulation of errors.

B Prompt Template

In Figure 5, we present the Chinese version example of the prompts utilized by Math-Critique and the corresponding English translations. In practical application, we employ the Chinese version of the prompts for all inquiries.

C Evaluation Settings

C.1 Evaluation Settings for Academic Datasets

Given our focus on evaluating zero-shot capabilities, traditional methods of answer verification fail to provide accurate assessments. Consequently, we have adapted our evaluation methodology as follows:

1. Extract the final sentence of the response (demarcated by a double newline character) or

Table 12: ChatGLM3-32B-Math Example (1)

Question	已知一个两位数的十位数字比个位数字大 2，两位数字的积比这个两位数小34，求这个两位数。
Question (Translate)	Given a two-digit number where the tens digit is 2 greater than the units digit, and the product of the digits is 34 less than the number itself, find this two-digit number.
Reference	设这个两位数为10x+y（x表示十位数，y表示个位数），根据题意可得：（1）x = y + 2 （十位数比个位数大2）（2）xy = 10x+y - 34 （两位数字的积比这个两位数小34）将（1）式代入（2）式得到：(y+2)y = 10(y+2)+y-34整理得到 $y^2 + 2y = 11y + 20 - 34$，整理得 $y^2 - 9y + 14 = 0$，解方程得到 y=7 或 y=2.所以x=7+2=9或x=2+2=4所以这个数为97或者42.答案：97或42。
Reference (Translate)	Let the two-digit number be $10x + y$ (where x is the tens digit and y is the units digit). According to the problem: (1) $x = y + 2$ (the tens digit is 2 greater than the units digit) (2) $xy = 10x + y - 34$ (the product of the digits is 34 less than the number itself). Substituting (1) into (2) gives: $(y + 2)y = 10(y + 2) + y - 34$, which simplifies to $y^2 + 2y = 11y + 20 - 34$, and further simplifies to $y^2 - 9y + 14 = 0$. Solving the equation yields $y = 7$ or $y = 2$. Thus, $x = 7 + 2 = 9$ or $x = 2 + 2 = 4$, making the number 97 or 42. Answer: 97 or 42.
GPT-4-0613	设这个两位数为10x+y，其中x为十位数，y为个位数。根据题意，我们有以下两个方程：x = y + 2 （十位数字比个位数字大2）xy = 10x + y - 34 （两位数字的积比这个两位数小34）将第一个方程代入第二个方程，我们得到：(y + 2)y = 10(y + 2) + y - 34解这个方程，我们得到y = 4。将y = 4代入第一个方程，我们得到x = 6。所以，这个两位数是64。
GPT-4-0613 (Translate)	Let the two-digit number be $10x + y$, where x is the tens digit and y is the units digit. According to the problem, we have two equations: $x = y + 2$ (the tens digit is 2 greater than the units digit) and $xy = 10x + y - 34$ (the product of the digits is 34 less than the number itself). Substituting the first equation into the second, we get: $(y + 2)y = 10(y + 2) + y - 34$. Solving this equation yields $y = 4$. Substituting $y = 4$ back into the first equation gives $x = 6$. Therefore, the two-digit number is 64.
ChatGLM3-32B-SFT	设这个两位数为 $10a + b$，其中 a 是十位数字，b 是个位数字。根据题意，我们有以下两个条件：1. 十位数字比个位数字大 2，即 $a = b + 2$ 2. 两位数字的积比这个两位数小 34，即 $ab < 10a + b - 34$ 现在我们来解这个两个方程。从第一个条件，我们得到 $a = b + 2$。将 a 替换进第二个条件中，我们得到：$b(b + 2) < 10(b + 2) - 34$ 展开并简化这个不等式：$b^2 + 2b < 10b + 20 - 34$ $b^2 + 2b < 7b - 14$ 移项得到：$b^2 - 5b + 14 < 0$ 这是一个一元二次不等式，我们可以求解这个不等式得到 b 的范围。因式分解得到：$(b - 1)(b - 14) < 0$ 根据一元二次不等式的性质，我们知道 b 应该在两个根之间，即：$1 < b < 14$ 但是由于 $a = b + 2$，且 a 是一个十位数（即 $1 \leq a \leq 9$），所以 b 必须小于 9。因此 b 的可能取值为 $2, 3, 4, 5, 6, 7, 8$。现在我们来检查 b 的每一个值，并找出对应的 a 和两位数。当 $b = 2$ 时，$a = 0$，这不满足 a 为十位数的要求；当 $b = 3$ 时，$a = 1$，这也不满足 a 为十位数的要求；...当 $b = 7$ 时，$a = 5$，这满足要求，并且 a 和 b 的积是 35，确实比 35 小 34，满足题目的要求。所以这个两位数是 57。
ChatGLM3-32B-SFT (Translate)	Let the two-digit number be $10a + b$, where a is the tens digit and b is the units digit. According to the problem, we have two conditions: 1. The tens digit is 2 greater than the units digit, i.e., $a = b + 2$ 2. The product of the digits is 34 less than the number itself, i.e., $ab < 10a + b - 34$ Now let's solve these two equations. From the first condition, we get $a = b + 2$. Substituting a into the second condition, we get: $b(b + 2) < 10(b + 2) - 34$ Expanding and simplifying this inequality gives: $b^2 + 2b < 10b + 20 - 34$ $b^2 + 2b < 7b - 14$ Rearranging gives: $b^2 - 5b + 14 < 0$ This is a quadratic inequality, which we can solve to find the range for b. Factoring gives: $(b - 1)(b - 14) < 0$ According to the properties of quadratic inequalities, we know b should be between the two roots, i.e., $1 < b < 14$. However, since $a = b + 2$ and a is a tens digit ($1 \leq a \leq 9$), b must be less than 9. Thus, the possible values for b are $2, 3, 4, 5, 6, 7, 8$. Now we check each value of b to find the corresponding a and the two-digit number. When $b = 2$, $a = 0$, which does not meet the requirement for a to be a tens digit; when $b = 3$, $a = 1$, which also does not meet the requirement; ... when $b = 7$, $a = 5$, which meets the requirement, and the product of a and b is 35, indeed less than 35 by 34, meeting the problem's requirement. Therefore, the two-digit number is 57.
ChatGLM3-32B-Math	设这个两位数为 $10a + b$，其中 a 是十位数字，b 是个位数字。根据题目条件，我们有以下两个方程：1. 十位数字比个位数字大 2，即 $a = b + 2$。2. 两位数字的积比这个两位数小 34，即 $ab = (10a + b) - 34$。我们可以用第一个方程代入第二个方程来解这个问题。将 $a = b + 2$ 代入第二个方程，得到：$b(b + 2) = 10(b + 2) + b - 34$ 展开并简化：$b^2 + 2b = 10b + 20 + b - 34$ $b^2 + 2b = 11b - 14$ 移项得到一个二次方程：$b^2 - 9b + 14 = 0$ 这是一个二次方程，我们可以通过分解因式来解它：$(b - 7)(b - 2) = 0$ 所以，b 的值可以是 7 或者 2。如果 $b = 7$，则 $a = b + 2 = 9$，所以这个两位数是 97。如果 $b = 2$，则 $a = b + 2 = 4$，所以这个两位数是 42。因此，这个两位数是 42 或 97。
ChatGLM3-32B-Math (Translate)	Let the two-digit number be $10a + b$, where a is the tens digit and b is the units digit. According to the problem conditions, we have two equations: 1. The tens digit is 2 greater than the units digit, i.e., $a = b + 2$. 2. The product of the digits is 34 less than the number itself, i.e., $ab = (10a + b) - 34$. We can solve this problem by substituting the first equation into the second. Doing so gives: $b(b + 2) = 10(b + 2) + b - 34$ Expanding and simplifying yields: $b^2 + 2b = 10b + 20 + b - 34$ $b^2 + 2b = 11b - 14$ Rearranging into a quadratic equation gives: $b^2 - 9b + 14 = 0$ This quadratic equation can be solved by factoring: $(b - 7)(b - 2) = 0$ Thus, b can be either 7 or 2. If $b = 7$, then $a = b + 2 = 9$, making the two-digit number 97. If $b = 2$, then $a = b + 2 = 4$, making the two-digit number 42. Therefore, the two-digit number could be either 42 or 97.

Table 13: ChatGLM3-32B-Math Example (2)

<table>
<tr><td>Question</td><td>两个相互独立的事件A，B都不发生的概率为1/9，A发生而B不发生的概率与A不发生B发生的概率相等，则P(A)等于?</td></tr>
<tr><td>Question (Translate)</td><td>Given two independent events A and B, where the probability of neither A nor B occurring is 1/9, and the probability of A occurring without B equals the probability of B occurring without A, what is P(A)?</td></tr>
<tr><td>Reference</td><td>由题意的：$P(A\bar{B}) = P(B\bar{A})$,即P(A)(1-P(B))=P(B)(1-P(A)) 所以P(A)=P(B) 又$P(\bar{A}\bar{B}) = \frac{1}{9}$ 所以$P(\bar{A}) = \frac{1}{3}$ 所以$P(A) = \frac{2}{3}$</td></tr>
<tr><td>Reference (Translate)</td><td>From the problem statement: P(A and not B) = P(B and not A), i.e., P(A)(1-P(B)) = P(B)(1-P(A)). Thus, P(A) = P(B). Also, P(neither A nor B) = 1/9. Therefore, P(not A) = 1/3, leading to P(A) = 2/3.</td></tr>
<tr><td>GPT-4-0613</td><td>设A发生的概率为p，B发生的概率为q，则A，B都不发生的概率为(1-p)(1-q)=1/9，A发生而B不发生的概率为p(1-q)，A不发生B发生的概率为(1-p)q。由题意知，p(1-q)=(1-p)q，解得p=q。 将p=q代入(1-p)(1-q)=1/9，得$(1-p)^2 = 1/9$，解得p=2/3或p=-1/3。由于概率不能为负，所以p=2/3，即P(A)=2/3。</td></tr>
<tr><td>GPT-4-0613 (Translate)</td><td>Let the probability of A occurring be p, and the probability of B occurring be q. Then, the probability of neither A nor B occurring is (1-p)(1-q) = 1/9. The probability of A occurring without B is p(1-q), and the probability of B occurring without A is (1-p)q. From the problem, p(1-q) = (1-p)q, leading to p = q. Substituting p = q into (1-p)(1-q) = 1/9 gives $(1-p)^2 = 1/9$, resulting in p = 2/3 or p = -1/3. Since probability cannot be negative, p = 2/3, i.e., P(A) = 2/3.</td></tr>
<tr><td>ChatGLM3-32B-SFT</td><td>设事件A发生的概率为P(A)，事件B发生的概率为P(B)。已知A和B相互独立，那么事件A不发生而B发生的概率为P(B)(1-P(A))，事件A发生而B不发生的概率为P(A)(1-P(B))。 根据题目条件，A发生而B不发生的概率与A不发生B发生的概率相等，即： P(A)(1-P(B)) = P(B)(1-P(A)) 由于A和B相互独立，我们可以将上式简化为： P(A) - P(A)P(B) = P(B) - P(B)P(A) 由于A和B的概率都是非负的，上式意味着P(A) = P(B)。但独立事件的概率相等并不一定意味着A和B是同一个事件，因为它们可以有相同的或不同的概率但独立性保持不变。 现在，我们知道了两个事件独立，且都不发生的概率为1/9。这意味着： P(A) * P(B) = 1/9 由于P(A) = P(B)，我们可以将这个等式写成： $P(A)^2 = 1/9$ 因为P(A)是概率，所以它必须在0和1之间。唯一满足这个等式的概率值是： $P(A) = æ(1/9) = 1/3$ 因此，P(A) = 1/3。既然P(A) = P(B)，那么P(B)也等于1/3。</td></tr>
<tr><td>ChatGLM3-32B-Math</td><td>设事件A发生的概率为P(A)，事件B发生的概率为P(B)。根据题意，我们可以列出以下关系： 1. 事件A和事件B都不发生的概率为1/9，即：

$$P(\text{A不发生且B不发生}) = P(\text{A不发生}) \times P(\text{B不发生}) = (1-P(A)) \times (1-P(B)) = \frac{1}{9}$$

2. 事件A发生而事件B不发生的概率与事件A不发生而事件B发生的概率相等，即：

$$P(\text{A发生且B不发生}) = P(\text{A不发生且B发生})$$

根据概率的性质，我们有：

$$P(\text{A发生且B不发生}) = P(A) \times (1-P(B))$$

$$P(\text{A不发生且B发生}) = (1-P(A)) \times P(B)$$

由于这两个概率相等，我们可以得到：

$$P(A) \times (1-P(B)) = (1-P(A)) \times P(B)$$

现在我们有两个方程： 1. $(1-P(A)) \times (1-P(B)) = \frac{1}{9}$ 2. $P(A) \times (1-P(B)) = (1-P(A)) \times P(B)$ 我们可以通过这两个方程来解出P(A)和P(B)的值。 首先，从第二个方程中我们可以得到：

$$P(A) - P(A) \times P(B) = P(B) - P(A) \times P(B)$$

$$P(A) = P(B)$$

这意味着事件A和事件B发生的概率相等。 现在，我们用这个结论来解第一个方程：

$$(1-P(A)) \times (1-P(A)) = \frac{1}{9}$$

$$(1-P(A))^2 = \frac{1}{9}$$

取平方根得到：

$$1-P(A) = \frac{1}{3} \quad \text{或} \quad 1-P(A) = -\frac{1}{3}$$

由于概率值不能为负，我们只取正值：

$$1-P(A) = \frac{1}{3}$$

$$P(A) = \frac{2}{3}$$

所以，事件A发生的概率P(A)等于2/3。</td></tr>
</table>

Table 14: ChatGLM3-32B-Math Example (3)

Question	The first term of an arithmetic sequence is 18. The sum of the first six terms of the sequence is equal to the sum of the first seven terms.Show that the sum of the first 13 terms of the sequence is 0 and calculate the 13th term of the sequence.
Reference	If the sum of the first six terms of a sequence is equal to the sum of the first seven terms then the seventh term must be 0. Let d be the common difference of the sequence. Now 18 + 6d = 0 and so d = -3. a13 = 18 + 12*(-3) = -18
ChatGLM3-32B-Math	Let's denote the common difference of the arithmetic sequence as d. The formula for the n-th term of an arithmetic sequence is $a_n = a_1 + (n-1)d$, where a_1 is the first term and n is the term number. Given that the first term $a_1 = 18$, the formula for the n-th term becomes $a_n = 18 + (n-1)d$. The sum S_n of the first n terms of an arithmetic sequence is given by the formula $S_n = \frac{n}{2}[2a_1 + (n-1)d]$. According to the problem, the sum of the first six terms (S_6) is equal to the sum of the first seven terms (S_7). We can write this relationship as: $S_6 = S_7$ $\frac{6}{2}[2 \cdot 18 + (6-1)d] = \frac{7}{2}[2 \cdot 18 + (7-1)d]$ $3[36 + 5d] = \frac{7}{2}[36 + 6d]$ $108 + 15d = \frac{7}{2} \cdot 36 + \frac{7}{2} \cdot 6d$ $108 + 15d = 126 + 21d$ Now, we solve for d: $15d - 21d = 126 - 108$ $-6d = 18$ $d = -3$ Now that we have the common difference $d = -3$, we can find the 13th term a_{13}: $a_{13} = 18 + (13-1) \cdot (-3)$ $a_{13} = 18 - 12 \cdot 3$ $a_{13} = 18 - 36$ $a_{13} = -18$ Now, let's find the sum of the first 13 terms S_{13} using the formula: $S_{13} = \frac{13}{2}[2 \cdot 18 + (13-1) \cdot (-3)]$ $S_{13} = \frac{13}{2}[36 - 12 \cdot 3]$ $S_{13} = \frac{13}{2}[36 - 36]$ $S_{13} = \frac{13}{2} \cdot 0$ $S_{13} = 0$ So, the sum of the first 13 terms of the sequence is indeed 0, and we've also calculated that the 13th term of the sequence is -18.

Table 15: ChatGLM3-32B-Math Error Case (1)

Question	The matrices $$\begin{pmatrix} 2 & a \\ -3 & -1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} -\frac{1}{16} & b \\ \frac{3}{16} & \frac{1}{8} \end{pmatrix}$$ are inverses. Enter the ordered pair (a, b).
Reference	$(6, -\frac{3}{8})$
ChatGLM3-32B-Math	Two matrices are inverses if their product is the identity matrix. The identity matrix for a 2x2 matrix is $$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$ So, we need to find the values of a and b such that $$\begin{pmatrix} 2 & a \\ -3 & -1 \end{pmatrix} \begin{pmatrix} -\frac{1}{16} & b \\ \frac{3}{16} & \frac{1}{8} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$ We can calculate the product as follows: $$\begin{pmatrix} 2 & a \\ -3 & -1 \end{pmatrix} \begin{pmatrix} -\frac{1}{16} & b \\ \frac{3}{16} & \frac{1}{8} \end{pmatrix} = \begin{pmatrix} 2(-\frac{1}{16}) + a(\frac{3}{16}) & 2b + a(\frac{1}{8}) \\ -3(-\frac{1}{16}) - 1(\frac{3}{16}) & -3b - 1(\frac{1}{8}) \end{pmatrix}.$$ This gives us the equations $2(-\frac{1}{16}) + a(\frac{3}{16}) = 1$, $2b + a(\frac{1}{8}) = 0$, $-3(-\frac{1}{16}) - 1(\frac{3}{16}) = 0$ $-3b - 1(\frac{1}{8}) = 1$ Simplifying, we get $-\frac{1}{8} + \frac{3a}{16} = 1$, $2b + \frac{a}{8} = 0$, $\frac{3}{16} - \frac{3}{16} = 0$, $-3b - \frac{1}{8} = 1$. From the third equation, we have $\frac{a}{8} = 2b$. Multiplying both sides by 8, we get $a = 16b$. Substituting into the first equation, we get $$-\frac{1}{8} + \frac{3(16b)}{16} = 1,$$ which simplifies to $3b = \frac{5}{8}$. Therefore, $b = \frac{5}{24}$. Substituting into the second equation, we get $$2(\frac{5}{24}) + \frac{16b}{8} = 0,$$ which simplifies to $b = -\frac{5}{16}$. This is consistent with our previous calculation, so $b = \frac{5}{24}$ is the correct value. Finally, substituting into the fourth equation, we get $$-3(\frac{5}{24}) - \frac{1}{8} = 1,$$ which simplifies to $a = \frac{1}{16}$. 9753 Therefore, $(a, b) = \boxed{\left(\frac{1}{16}, \frac{5}{24}\right)}$.

Table 16: ChatGLM3-32B-Math Error Case (2)

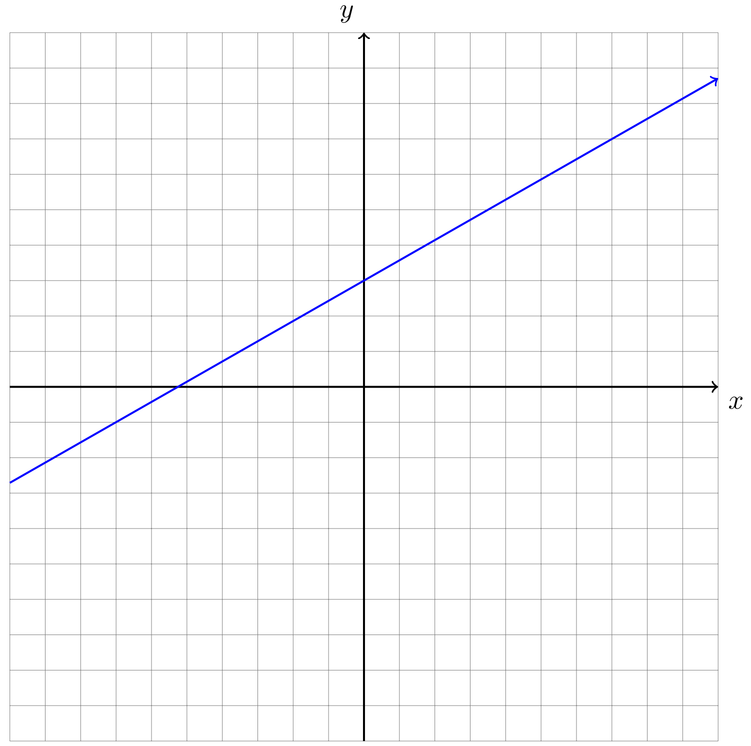
Question	n the graph below, each grid line counts as one unit. The line shown below passes through the point $(1001, n)$ (not shown on graph). Find n.  **(note: the graph was written in Asymptote code in original question)**
Reference	575
ChatGLM3-32B-Math	The line passes through the point $(1001, n)$, which means that when $x = 1001$, $y = n$. We can find the slope of the line by looking at the change in y divided by the change in x between two points on the line. If we consider the point $(999, 7)$ and the point $(1001, n)$, the change in x is $1001 - 999 = 2$ and the change in y is $n - 7$. Since the line passes through the point $(1001, n)$, we can say that the slope of the line is $\frac{n-7}{2}$. We know that the line also passes through the point $(0, -2.71)$, so we can use this point to find the slope of the line. The change in x is $0 - (-10) = 10$ and the change in y is $-2.71 - (-7) = 4.29$. Now we can set the two slopes equal to each other and solve for n: $$\frac{n-7}{2} = \frac{4.29}{10}$$ Cross multiplying gives: $$10(n - 7) = 2 \cdot 4.29$$ $$10n - 70 = 8.58$$ Adding 70 to both sides gives: $$10n = 78.58$$ Finally, dividing both sides by 10 gives: $$9754_{n} = 7.858$$ But since the grid lines count as one unit, we need to round n to the nearest whole number. 7.858 is closer to 8 than it is to 7, so $n = \boxed{8}$.

Table 17: ChatGLM3-32B-Math Error Case (3)

<table>
<tr><td>Question</td><td>In the year 2018 factory A produced goods worth 500 million forints, while factory B produced goods worth 400 million forints. According to their long-term plans, factory A will increase production by 5% each year, while factory B will increase production by 6% per year.
The above plans were published on an economy website. A debate started in the comment section of the article: one user claimed that the difference between the volumes of production between the two factories would get smaller and smaller in the next couple of years.
Calculate the missing values of the table below and show that the above claim is not true.
| Year | Production volume of factory A (million Ft) | Production volume of factory B (million Ft) |
——	————————————————	————————————————
2018	500	400
2019		
2020		
2021		|</td></tr>
<tr><td>Reference</td><td>| Year	Production volume of factory A (million Ft)	Production volume of factory B (million Ft)
——	————————————————	————————————————
2018	500	400
2019	525	424
2020	551.3	449.4
2021	578.8	476.4 | The difference between the production values in the given years: 525 – 424 = 101, 551.3 – 449.4 = 101.9 and 578.8 – 476.4 = 102.4 (million forints). As the difference between the production values is increasing (in the period in question), the statement is, in fact, false.</td></tr>
<tr><td>ChatGLM3-32B-Math</td><td>To calculate the production volumes for the years 2019, 2020, and 2021, we will apply the percentage increases to the production of each factory. For Factory A: - 2018 production: 500 million Ft - 5 For Factory B: - 2018 production: 400 million Ft - 6 Let's fill in the table: | Year	Production volume of factory A (million Ft)	Production volume of factory B (million Ft)
——	————————————————	————————————————
2018	500	400
2019	500 * 1.05 = 525	400 * 1.06 = 424
2020	525 * 1.05 = 551.25	424 * 1.06 = 448.24
2021	551.25 * 1.05 = 578.96	448.24 * 1.06 = 475.92 | Now, let's analyze the data. The difference in production between the two factories in 2018 was 100 million Ft (500 million - 400 million). - In 2019, the difference is 21 million Ft (525 million - 424 million). - In 2020, the difference is 23.31 million Ft (551.25 million - 448.24 million). - In 2021, the difference is 23.94 million Ft (578.96 million - 475.92 million). The difference is not getting smaller; in fact, it is increasing slightly each year. This is because the rate of increase for Factory A (5%) is lower than the rate for Factory B (6%). The claim made in the debate that the difference would get smaller and smaller is not true based on these figures.</td></tr>
</table>

prompt the model with "Therefore, the answer is".

2. For numeric standard answers, the response undergoes the following recognition process: fractions or decimals in LaTeX format, and regular fractions, decimals, or integers. We compute the numerical result and compare it with the standard answer. A discrepancy less than (1e-6) is deemed correct.

3. For standard answers that are strings (exclusive to the MATH dataset): identify the content following "xxx is" or an equality sign, or within $boxed\{\}$. The evaluation considers whether the normalized strings match.

For the English datasets GSM8k and MATH, our principle for selecting results is as follows:

1. If the model itself reports results on these two datasets, we choose the higher of either the zero-shot or few-shot results.

2. If the model does not report its results, we utilize the results reported by OpenCompass, again selecting the higher of the zero-shot or few-shot outcomes. This applies to models such as GPT-4-0613, GPT-3.5-Turbo-0613, Yi-6B-Chat, and Qwen-Chat-7B. Should there be any omissions from the above sources, results will not be reported.

For the Hungarian Exam results, we had annotators score them according to a grading standard, which is referenced from (Paster, 2023). It's noted that the reported results actually sum the scores of all questions, making the total score effectively 117. To align with the reported results, we adopted this scoring method as well.

With reproducibility in mind, all our results were obtained using a sampling temperature of 0 and setting the max-seq-length to 4096.

C.2 Evaluation Settings for 2023 Hungarian national high school finals in mathematics

For the Hungarian national high school finals in mathematics, we submit the model's answers to annotators for marking. For results of models not listed in (Paster, 2023), we score them based on the answers provided in (Paster, 2023) according to the scoring points. We sum the scores of all questions to present a total score. All annotations are carried out by two annotators; in case of inconsistency, a third annotator decides.

Considering the general situation of multiple models, we do not restrict the language used by the language models to answer the questions. Any language used to correctly answer is considered correct. Additionally, since most questions do not restrict the form of the answer, we stipulate that answers are deemed correct as long as they retain more than one decimal place accurately or are provided in fraction form.

D Additional Results

D.1 Subcategory Results of MathUserEval

In Table 18, we display the results for all subsets of MathUserEval. The reported results were evaluated by GPT-4-1106-Preview, with the evaluation method consistent with AlignBench. It is noted that GPT-4-0125-Preview and GPT-4-1106-Preview still occupy the leading positions. Except for Probability, the GLM4 model's total score and individual scores surpassed GPT-4-0613. Our GLM-Math-32B w/ DPO model performed exceptionally well in the Elementary category, exceeding GPT-4-0613, but a significant gap remains in Advanced mathematics. Our Self-Critique training method showed significant progress in MathUserEval, with an overall improvement of 24%.

D.2 Subcategory Results of Alignbench (Liu et al., 2023a)

Table 19 reports detailed results from the language capability subsection of AlignBench. Within this, we present the scores of our four models and have tested the results for Qwen-72B-Chat (Bai et al., 2023a), Claude-2 (Anthropic, 2023), and Yi-34B-Chat (Yi, 2023). Additional results are derived from the AlignBench paper, and the results for DeepSeek are taken from its report (DeepSeek-AI et al., 2024).

D.3 Additional Language Abilities

The paper utilized Alignbench and MTbench as representative general capability test sets for Chinese and English. These are among the most important general capability test sets for both languages. We further supplemented our analysis with tests on MMLU, Ceval, CMMLU, and ARC, famous language abilities evaluation benchmarks in Table 20, confirming no significant decline in our method across a broader range of general capability tests.

Table 18: Math-User-Result Result, GPT-4-1106-Preview-rated. All results were scored by GPT-4-1106-Preview, with the scoring method consistent with AlignBench. All Overall scores were calculated using the macro-average.

Model	Overall	Elementary					Advanced				
		Avg	algebra	calculate	geo.	tri.	Avg	calculus	discrete	linear.	Prob.
GPT-4-0125-Preview (OpenAI, 2023)	**5.79**	**5.26**	**5.04**	**7.63**	**3.98**	4.59	6.71	7.26	6.62	**5.48**	7.72
GPT-4-1106-Preview (OpenAI, 2023)	5.73	5.07	4.96	7.00	3.78	4.71	**6.81**	**7.39**	**6.96**	5.29	**7.91**
GLM-4	5.11	4.86	4.47	6.56	3.95	**4.74**	5.43	6.00	5.67	4.26	6.02
ChatGLM3-32B-SFT-2312 + RFT&DPO	4.23	4.01	3.88	5.41	2.90	3.99	4.59	5.22	4.76	3.38	5.20
GPT-4-0613 (OpenAI, 2023)	4.14	3.34	2.88	4.76	3.17	2.78	5.33	5.57	5.49	4.26	6.22
ChatGLM3-32B-SFT-2312 + RFT	4.01	3.86	3.84	5.37	2.57	3.77	4.26	4.72	4.69	2.98	4.89
Qwen-72B-Chat (Bai et al., 2023a)	3.87	3.99	3.96	4.81	3.83	3.34	3.67	4.54	3.71	2.84	3.65
GPT-3.5-Turbo-0613 (OpenAI, 2023)	3.42	3.04	2.81	4.07	2.23	3.26	4.07	4.83	4.38	3.26	3.91
ChatGLM3-32B-SFT-2312	3.39	3.35	3.35	4.51	2.51	3.11	3.44	4.04	4.38	2.41	3.13
Claude-2 (Anthropic, 2023)	3.29	2.63	2.35	3.63	2.20	2.53	4.35	4.56	4.53	3.29	5.28
DeepSeek-Chat-67B (DeepSeek-AI et al., 2024)	3.24	2.76	2.21	4.73	2.12	2.30	3.84	4.41	4.82	2.79	3.52
Yi-34B-Chat (Yi, 2023)	2.64	2.49	2.04	3.61	2.25	2.27	2.87	2.80	3.47	2.03	3.41

Table 19: Results of Alignbench (Liu et al., 2023a), Language Part.

Model	Language						
	Avg.	**Fund.**	**Chi.**	**Open.**	**Writ.**	**Role.**	**Pro.**
GPT-4-1106-Preview (OpenAI, 2023)	8.29	7.99	7.33	8.61	8.67	8.47	8.65
ChatGLM3-32B-SFT-2312 + RFT&DPO	7.80	7.14	6.90	8.37	8.41	8.09	7.90
GPT-4-0613 (OpenAI, 2023)	7.59	7.81	6.93	7.42	7.93	7.51	7.94
ChatGLM3-32B-SFT-2312 + RFT	7.43	6.37	6.95	8.03	7.71	7.97	7.54
ChatGLM3-32B-SFT-2312	7.38	6.84	7.02	8.08	7.37	7.70	7.27
Qwen-72B-Chat (Bai et al., 2023a)	7.29	6.63	7.31	7.24	7.29	7.59	7.71
DeepSeek-67B-Chat (DeepSeek-AI et al., 2024)	7.11	7.12	6.52	7.58	7.20	6.91	7.37
GPT-3.5-Turbo-0613 (OpenAI, 2023)	6.82	6.71	5.81	7.29	7.03	7.28	6.77
Claude-2 (Anthropic, 2023)	6.78	6.87	6.24	7.08	6.36	6.85	7.31
Yi-34B-Chat (Yi, 2023)	6.18	4.32	6.05	7.37	6.00	6.30	7.06

[1] The ChatGLM3-32B-SFT-2312 is a newer version of the ChatGLM series and not identical to the model discussed in (Hou et al., 2024), despite sharing the same model size.

Table 20: Performance comparison across different language abilities evaluation datasets.

	MMLU	CEVAL	CMMLU	ARC-E	ARC-C
ChatGLM3-32B-SFT-2312	0.593	0.602	0.656	0.914	0.777
ChatGLM3-32B-SFT-2312 + RFT	0.659	0.751	0.793	0.974	0.924
ChatGLM3-32B-SFT-2312 + RFT & DPO	0.665	0.769	0.793	0.974	0.922

Notably, all additional training data we incorporated were math-related, with over 90% being in Chinese. Hence, maintaining the English general capability performance in MTbench was in line with our objectives, as we did not claim our model to be the superior English language model, nor were we certain of language ability enhancement without relevant data addition.

Regarding mathematical capabilities, our RFT and DPO versions outperformed Qwen-math in all but one math test set, Cmath, and showed an average improvement of +9.4% and +16.4% across all math benchmarks. Specifically, within the Math-UserEval categories, we surpassed Qwen-math in 6 out of 8 categories. Additionally, our model is only half the size of Qwen-math.

D.4 Effectiveness of Math-Critique

During the manual annotation process, we collected a test set of 800 questions, manually marked for correctness and procedures, forming a four-category test set. The output results of Math-

Critique were mapped to these four categories as per the instructructions.

We validated the effectiveness of Math-Critique through empirical experiments with two evaluation methods: the accuracy of directly scoring to judge correct/incorrect results and the accuracy of judging our defined four categories. Test sets were extracted from Chinese junior and senior high school exam questions and MATHUSEREVAL, with experts annotated correct judgments.

The results in Table 7 indicate that our Math-Critique-32B model significantly surpasses GPT-3.5-Turbo in both judgment accuracy and correlation coefficients compared to human annotations and is essentially on par with GPT-4-0613.

D.5 Comparison of Self-improvement Algorithms

To better compare different self-improvement algorithms, we conducted experiments using the LLaMA3-8B-instruct model as the baseline. We sampled the same training data size for the STAR method and our approach, training for three epochs. As shown in Table 21, our method improved from 64.5 to 75.3, outperforming the RFT and STAR methods. We also present the scores for seven additional test datasets in MAmmoTH (Yue et al., 2023) in Table 22. The scores for GPT-4 and MAmmoTH-70B are taken from the original paper. It can be seen that our method achieves an average improvement from 61.6 -> 72.5 across all seven OOD test datasets.

In the RFT setting, we applied RFT to the academic subset of our corpus, specifically GSM8k and Math datasets. In the STAR setting, we used the rationalization method described in the STaR paper, where reference answers were used as prompts to train on our corpus. Upon reviewing the composition of our training corpus, which included GSM8k and MATH training sets, as well as Chinese test papers and internet-sourced problems, we observed that in datasets with high-quality reference answers like GSM8k and MATH, the STaR method with rationalization achieved notable improvements, though still slightly lower than our method. However, the STaR method with rationalization showed minimal or no improvement in test sets with lower-quality reference answers. This may be because the model has not yet developed the ability to derive accurate solutions from low-quality reference answers.

D.6 Performance on Question Length

As shown in Table 23, we followed MetaMath's approach to performing an ablation study on the question length for GSM8k and MATH. While long questions remain more challenging for the model both before and after applying our method, it is evident that our method enhances performance across questions of varying lengths.

E Comparsion with other work

E.1 Self-Critique Pipeline

While the thought of self-critique has been proposed, it has not been explored sufficiently in LLMs' math problem-solving improvement and real-world large-scale deployment. We provide a detailed comparison in Table 2. Thus, our unique contributions in this work lie in realizing the idea in a real-world massively deployed LLM via several novel designs, including:

Application Domain of Math: While previous works have broadly focused on general language capabilities, our research is the first to focus on improving mathematical problem-solving skills using self-critique methods without affecting general language abilities.

Unified and Novel Approach for Data Selection and Reward Construction: We introduce a novel approach by uniformly using Math-Critique to select Supervised Fine-Tuning (SFT) data and construct reward signals for Reinforcement Learning (RL). This method offers higher critique precision in the mathematics domain, different from the varied strategies employed in earlier studies.

Independence from External Support: Our approach is the first to achieve both: 1. Independence from human input and more powerful external models during training, and 2. The generation of training data and reward signals by a model that has been fine-tuned from the same foundational model. This emphasizes our method's unique capability for self-sufficiency and self-improvement.

E.2 MathUserEval Benchmark

The MathUserEval benchmark is designed to address the limitations of traditional mathematical benchmarks through:

Data Source: Drawing from real-world user scenarios and university exams, offering a diversity closer to actual use.

Difficulty and Domain: This course covers a wide range of mathematics, from elementary to

Table 21: Performance comparison across different self-improvement algorithms.

Models	**Chinese Test Set**			**English Test Set**			**AVG-academic**
	MathUserEval	Math23k	Ape210k500	Cmath	GSM8k	MATH	
LLaMA3-8B-instruct	24.8	63.4	73.6	74.3	77.6	33.6	64.5
LLaMA3-8B-instruct+RFT	22.8	57.7	65.4	69.2	79.2	35.2	61.3
LLaMA3-8B-instruct+STAR	23.2	65.1	72.8	68.3	83.7	40.3	66.0
LLaMA3-8B-instruct+**OURS**	33.9	80.9	84.2	83.5	84.2	43.7	75.3

Table 22: Performance comparison across different self-improvement algorithms on OOD test sets.

AVG	**AQuA**	**NumGLUE**	**SVA**	**Mat**	**Sim**	**SAT**	**MMLU**	**AVG**
GPT-4	-	72.6	-	97.0	-	-	95.0	-
MAmmoTH-70B	64.6	65.0	74.4	82.4	55.6	51.4	66.4	56.7
LLaMA3-8B-instruct	61.6	57.9	62.1	82.5	51.5	59.9	65.0	52.3
LLaMA3-8B-instruct+STAR	70.0	60.2	75.3	90.3	56.8	81.5	65.5	60.4
LLaMA3-8B-instruct+**OURS**	72.5	67.7	73.6	90.6	58.8	85.2	70.0	61.8

Table 23: Performance comparison across different question lengths on GSM8k and MATH.

Question length	**GSM8k**		**MATH500**	
	ChatGLM-32B-SFT	ChatGLM Math-32B	ChatGLM-32B-SFT	ChatGLM Math-32B
short	81.55%	89.52%	42.17%	57.23%
medium	77.45%	85.19%	24.10%	39.16%
long	68.71%	73.24%	20.83%	26.79%

university level, including computation, equations, calculus, probability, linear algebra, and discrete math.

Evaluation Method: Our evaluation caters to all forms of mathematical questions besides traditional multiple-choice or exact-match fill-in-the-blanks, enabling accurate assessment of complex expressions and proof questions.

Table 24: Data distribution of Math-Critique training data.

Label Type	Size
Annotation	6500
CritiqueLLM label	8800
Total	15300

F Data Collection

This section introduces the specific composition of our training data. Aside from simulated dialogue annotations in our data collection process, all annotators were from the crowdsourcing team, mostly undergraduate or higher-level students majoring in science and engineering from China. Annotators were clearly informed about the data usage and received fair compensation. All annotated data underwent a second review to ensure accuracy and the absence of any ethnic issues.

F.1 Training Data for Math-Critique

Most of the data was annotated using results from CritiqueLLM during the Math-Critique training, but some data was annotated manually. For this portion, we employed a crowd-sourced annotation team. After informing them that their annotations would be used for training or publishing purposes, we paid them an hourly wage based on the local average. All of our annotators possess at least a bachelor's degree to ensure they can correctly understand the requirements of the tasks. We require annotators to provide ratings that meet Math-Critique standards based on the problems, reference answers, and model responses. Additionally, they need to rewrite the feedback content according to these criteria. Data details are shown in Table 24.

Table 25: Distribution of the source dataset of training data for Critique RFT.

Source Dataset	Size
Primary	28597
Junior High	21303
Senior High	18917
College	4001
MetaMath	6291
Simulated Dialog	37653
Total	116762

Table 26: Distribution of the source dataset of training data for Critique DPO.

Source Dataset	Size
Primary	23485
Junior High	11304
Ssenior High	11026
College	3801
Simulated Dialog	12741
Total	62357

F.2 Training Data for Critique RFT

In Table 25, We provided the data proportions used for RFT training. Most of the data comes from primary and secondary school math problems. We also supplemented the dataset with MetaMath questions to include English data and Simulated Dialog from some of our collaborating annotators. This part of the data involved inviting a group of annotators, including middle school and university students, to provide the model with math problems they encountered daily. In total, 116k math instruction data points went through the abovementioned screening process.

F.3 Training Data for Critique DPO

As shown in Table 26, after filtering our DPO data with RFT model responses and Math-Critique, 62k entries remained for DPO training. It was observed that the number of problems from primary school, middle school, and Simulated Dialog decreased the most. At this stage, we removed all data originating from academic datasets because we found that including these data did not yield any benefits and significantly affected the results of the non-academic datasets.

F.4 References of MathUserEval Dataset

The standard answers for the MathUserEval dataset were all written by our crowdsourced annotation team. During the annotation process, annotators had access to example solutions from several advanced models for reference. However, annotators were required to solve the problems independently to produce the standard reference answers. All annotators in this task hold at least a bachelor's degree in mathematics or a related field.