MathAttack: Attacking Large Language Models towards Math Solving Ability

Zihao Zhou[1 2], **Qiufeng Wang**[1 *], **Mingyu Jin**[3], **Jie Yao**[1 2]
Jianan Ye[1 2], **Wei Liu**[4], **Wei Wang**[1], **Xiaowei Huang**[2], **Kaizhu Huang**[5]
[1]School of Advanced Technology, Xi'an Jiaotong-Liverpool University
[2]University of Liverpool
[3]Northwestern University
[4]ShanghaiTech University
[5]Duke Kunshan University
zihao.zhou@liverpool.ac.uk, Qiufeng.Wang@xjtlu.edu.cn

Abstract

With the boom of Large Language Models (LLMs), the research of solving Math Word Problem (MWP) has recently made great progress. However, there are few studies to examine the robustness of LLMs in math solving ability. Instead of attacking prompts in the use of LLMs, we propose a **MathAttack** model to attack MWP samples which are closer to the essence of robustness in solving math problems. Compared to traditional text adversarial attack, it is essential to preserve the mathematical logic of original MWPs during the attacking. To this end, we propose logical entity recognition to identify logical entries which are then frozen. Subsequently, the remaining text are attacked by adopting a word-level attacker. Furthermore, we propose a new dataset **RobustMath** to evaluate the robustness of LLMs in math solving ability. Extensive experiments on our **RobustMath** and two another math benchmark datasets GSM8K and MultiAirth show that **MathAttack** could effectively attack the math solving ability of LLMs. In the experiments, we observe that (1) Our adversarial samples from higher-accuracy LLMs are also effective for attacking LLMs with lower accuracy (e.g., transfer from larger to smaller-size LLMs, or from few-shot to zero-shot prompts); (2) Complex MWPs (such as more solving steps, longer text, more numbers) are more vulnerable to attack; (3) We can improve the robustness of LLMs by using our adversarial samples in few-shot prompts. Finally, we hope our practice and observation can serve as an important attempt towards enhancing the robustness of LLMs in math solving ability. The code and dataset is available at: https://github.com/zhouzihao501/MathAttack.

Introduction

Solving Math Word Problem (MWP) aims to infer a final answer from the natural language description of a math problem (Wang, Liu, and Shi 2017). With the boom of Large Language Models (LLMs), the research of solving MWP has recently made great progress (Qiao et al. 2022; Uesato et al. 2022; Chang et al. 2023). Most of them work on prompt engineering to improve math solving ability of LLMs (Wei et al. 2022; Zhou et al. 2023a; Kojima et al. 2022; Chen et al. 2022; Fu et al. 2023b; Wang et al. 2023c; Yao et al. 2023), and LLMs (e.g., ChatGPT) can provide correct reasoning process and the final answer for simple math word problems. Subsequently, they have been progressively applied in the field of intelligence education (Macina et al. 2023; Wang and Demszky 2023; Wang et al. 2023d; Handa et al. 2023). Therefore, it becomes essential to examine the robustness of LLMs in math solving ability, but this has not attracted much attention so far. To the best of our knowledge, there are only a few works (Zhu et al. 2023; Wang et al. 2023a) to evaluate the robustness of LLMs through attacking prompts (Figure 1(b)). By comparing to prompt-attack, we argue that attacking MWP samples themselves is more direct to reflect the robustness of LLMs in math solving ability, like Figure 1(c).

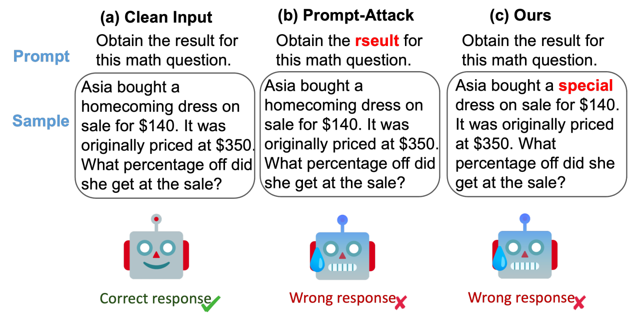

Figure 1: Different input of Large Language Models (LLMs). (a) Clean input, (b) Adversarial sample generated by Prompt-Attack (Zhu et al. 2023; Wang et al. 2023a), (c) Adversarial sample generated by our MathAttack.

On the other hand, general text adversarial attack has made great progress (Li et al. 2019, 2020; Ye et al. 2022; Qian et al. 2022). This task aims to generate an adversarial text x' that is semantically similar to the original text x, while victim model f can correctly classify x but incorrectly classify x' (Jin et al. 2020; Xu et al. 2020). However, it tends to change mathematical logic by directly applying such techniques of general text adversarial attack. For example, if the word 140 in Figure 1(c) is modified to another number, the mathematical logic will be changed and the original ground-truth will be no longer the correct answer. Therefore, it is essential to preserve the mathematical logic of MWPs, which

*Corresponding author

makes MWP adversarial attack more challenging.

To preserve the mathematical logic of MWPs, we propose **MathAttack** for attacking the math solving ability of large language models. Figure 2 shows an overview of our MathAttack. We first recognize logical entities, altering these logical entities easily leads to changing the mathematical logic of math word problems. Then we freeze the logical entities, preventing the attacker from modifying logical entities. Finally we attack the LLMs utilizing word-level attacker (Li et al. 2020) while not changing those frozen logical words. With the help of MathAttack and manual check, we propose a new dataset **RobustMath**, which consists of 300 high-quality MWP adversarial samples and could measure the robustness of LLMs' math solving ability.

Extensive experiments on our proposed **RobustMath** dataset and another two math benchmark datasets GSM8K (Cobbe et al. 2021) and MultiAirth (Roy and Roth 2015) show that our **MathAttack** could effectively attack the math solving ability of LLMs. As far as we know, most works (Zhu et al. 2023; Wang et al. 2023a) focus the robustness of LLMs in general tasks, there are not any comprehensive study on the robustness of LLMs in math solving ability. To this end, we conduct a a serious of analysis in the experiments and observe the following three points: (1) Transferability of attacking samples. Adversarial samples generated from higher-accuracy LLMs are also effective for attacking LLMs with lower accuracy (e.g., transfer from larger to smaller-size LLMs, or from few-shot to zero-shot prompts); (2) Complex MWPs (such as more solving steps, longer text, more numbers) are more vulnerable to attack; (3) We can improve the robustness of LLMs by using our attacking samples in few-shot prompts.

In summary, our contributions are as follows:

- In this paper, we make a first attempt to attack MWP samples to examine the robustness of LLMs in math solving ability.
- We propose **MathAttack** for attacking the math solving ability of LLMs, including Logical Entity Recognition, Freezing Logical Entity and text Attack.
- We propose a new dataset **RobustMath** by adopting MathAttack and manual check. It consists of 300 high-quality MWP adversarial samples and could measure the robustness of LLMs' math solving ability.
- Extensive experiments show that MathAttack could effectively attack the math solving ability of LLMs. Through the exhaustive analysis, we obtain three findings for the robustness of LLMs in math solving ability.

Related Work

MWP Solver Recent proposals intend to solve the problem by using sequence or tree generation models. (Wang, Liu, and Shi 2017) presents a sequence-to-sequence (seq2seq) approach to generate the mathematical equation. (Xie and Sun 2019) propose a goal-driven tree-structured (GTS) model to generate the equation tree. This sequence-to-tree approach significantly improves the performance over the traditional seq2seq approaches. (Zhang et al. 2020) adopt a graph-to-tree approach to model the quality relations using graph convolutional networks (GCN). Previous studies (Patel, Bhattamishra, and Goyal 2021; Zhou et al. 2023b; Yao, Zhou, and Wang 2023) indicate these MWP solvers rely on shallow heuristics to generate equations. With the boom of Large Language Models (LLMs) and the proposal of chain-of-thought (Wei et al. 2022), the math solving ability of the model has recently made great progress. Many research works on prompt engineering to improve math solving ability (Zhou et al. 2023a; Kojima et al. 2022; Chen et al. 2022; Fu et al. 2023b; Wang et al. 2023c; Yao et al. 2023), they are capable of effortlessly solving simple MWPs, and LLMs are gradually being incorporated in the field of intelligent education (Ji, Han, and Ko 2023; Macina et al. 2023; Wang and Demszky 2023; Wang et al. 2023d; Handa et al. 2023). In this context, examining the robustness of LLMs in math solving ability becomes essential. In this work, we make a first attempt to examine this robustness issue by attacking MWP samples.

Large Language Models Attack Previous proposals have already tried to evaluate the robustness of large language models (Zhuo et al. 2023; Shi et al. 2023). (Wang et al. 2023b) makes the first attempt to systematically evaluate the robustness of LLMs by using robust datasets. Recently, some works propose to address this issue by attacking prompts (Wang et al. 2023a; Zhu et al. 2023). (Wang et al. 2023a) introduces the ICL attack based on TextAttack, which aims to manipulate the prompt only without altering the input. (Zhu et al. 2023) presents PromptBench, a robustness benchmark specifically designed to evaluate the robustness of LLMs against adversarial prompts. Our work differs from theirs in two main aspects: (1) We specifically focus on attacking the MWP sample itself, which provides a more direct approach and fills the gap of non-prompt attacks on LLMs. (2) Their works target general tasks, lacking a comprehensive analysis of the robustness in math solving ability.

MWP Attack For the MWP solvers, previous works generate some MWP adversarial examples by rule-based methods like reordering the problem description (Kumar, Maheshwary, and Pudi 2021; Patel, Bhattamishra, and Goyal 2021). However, with the development of LLMs, the semantic and logical capabilities of the model have been enhanced, rendering these adversarial examples ineffective. Adversarial MWP sample datasets **SVAMP** (Patel, Bhattamishra, and Goyal 2021) can be solved well by LLMs like ChatGPT. In this paper, we attack MWP samples of LLMs for the first time and propose a new dataset **RobustMath** to evaluate the robustness of math solving ability of LLMs. It consists of adversarial examples generated by MathAttack, utilizing simple MWPs from GSM8K and MultiAirth as seed data.

Methodology

Problem Formulation

Suppose we have a text x with n words $x = [w_1, w_2, ..., w_n]$ whose ground truth label is y. We call x' an adversarial example when x' can make the victim model f wrong prediction but original correct prediction ($f(x) = y$), i.e.,

$$f(x') \neq f(x). \tag{1}$$

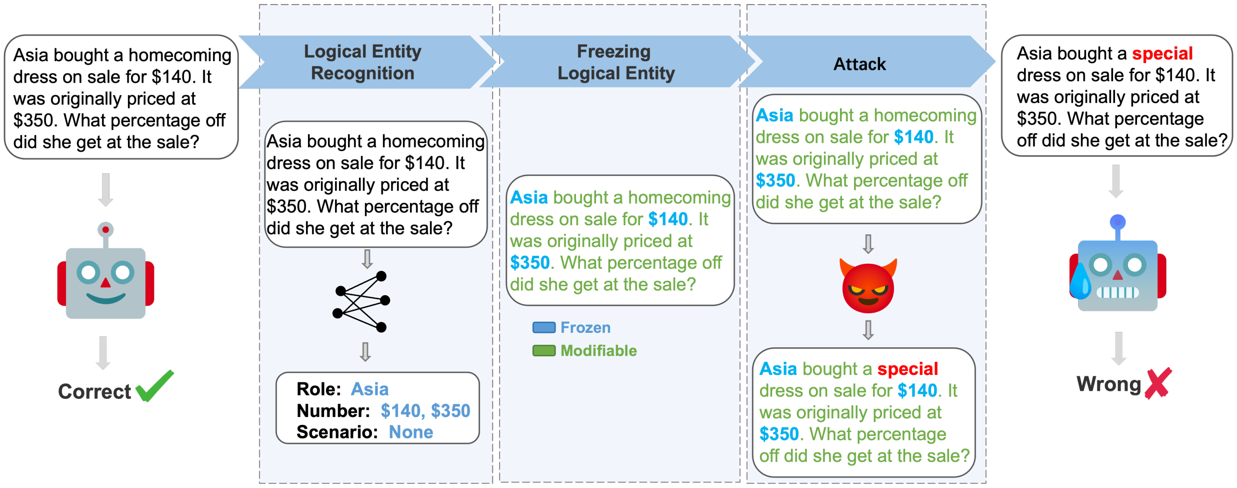

Figure 2: The overview of MathAttack. First, we utilize an NER model to identify logical entities. Then we freeze the logical entities, preventing the attacker from modifying them. Finally, we utilize word-level attacker to attack the LLMs while not changing those frozen logical entities.

Compared to traditional text attack, math word problem attack need to preserve the mathematical logical L of text sample x, it is defined as:

$$L\left(x'\right) = L\left(x\right). \tag{2}$$

The goal of the attack task is to generate an adversarial example x^* among all x'. Since text data consists of discrete words whose change can be perceived by humans, we always want the optimized adversarial example x^* to be semantically closest to the original text sample x. Thus, the objective function of this task can be defined as follows:

$$x^* = \underset{x'}{\operatorname{argmax}}\, \mathcal{G}\left(x, x'\right), s.t. f\left(x'\right) \neq f\left(x\right), L\left(x'\right) = L\left(x\right), \tag{3}$$

where $\mathcal{G}\left(x, x'\right)$ denotes the semantic similarity between x and x'. In this paper, f is the large language model and we follow the black-box setting.

The Proposed MathAttack

Overview Figure 2 shows an overview of MathAttack. We firstly recognize logical entities. Altering these logical entities easily leads to changing the logic of math word problems. Then we freeze the logical entities, preventing the attacker from modifying logical entities. Finally we attack the LLMs utilizing word-level attacker while not changing those frozen logical entities.

Logical Entity Recognition Logical entities are crucial components that constitute logic in math word problems (Kumar, Maheshwary, and Pudi 2022; Li et al. 2022). In order to preserve the logic of a math word problem, it is indispensable to define and identify which entities as logical entities. In this paper, we define the following three types of entities as logical entities. (1) **Role Entity**: It includes person entity (e.g, *Asia* in Figure 2). (2) **Number Entity**: It includes quantity (e.g, *$140* in Figure 2), cardinal number and ordinal number. (3) **Scenario Entity**: It includes time entity and location entity. Altering these environmental factors is easy to change the logic of math word problems too.

Then we employ Named Entity Recognition (NER) model to identify them:

$$I_{ro} = NER_{ro}\left(x\right), \tag{4}$$

$$I_{num} = NER_{num}\left(x\right), \tag{5}$$

$$I_{sce} = NER_{sce}\left(x\right), \tag{6}$$

where I_t is a word index set if the word belongs to logical entity type t. The symbols ro, num and sce represent the Role, Number and Scenario Entity respectively. We utilize Spacy [1] as our NER model.

Freezing Logical Entity It tends to break the original logic of MWP by altering logical entities during the attack process. To this end, we freeze all logical entities in order to prohibit attackers from modifying them:

$$I_f = I_{name} \cup I_{num} \cup I_{sce}, \tag{7}$$

where I_f denotes the frozen word index set.

Attack Text attackers are generally classified into three types: char-level, word-level and sentence-level. In MathAttack, We choose word-level attacker because the char-level attacker can distort the semantic meaning of words (like Figure 1(b)) and sentence-level attacker are prone to disrupting the mathematical logic of MWP. The attack process of word-level attacker primarily entails two steps: finding vulnerable words and words replacement.

In order to find vulnerable words, it is necessary to determine which words are significant. Specifically, we first sequentially mask all modifiable words to form new sentences. Afterward, we predict each new sentence to get the drop in

[1] https://spacy.io/

the probability of the correct answer. The more it drops, the more important the word is. It is defined as:

$$x_i^{mask} = [w_1, w_2, , w_{i-1}, mask, w_{i+1}, ..., w_n], i \notin I_f, \tag{8}$$

$$a_i = prob\,(f\,(x)) - prob\left(f\left(x_i^{mask}\right)\right), \tag{9}$$

where a_i is the important score of x_i and $prob$ is the function to get the probability of the correct answer. After that, we can get the important scores list $a = [a_1, a_2, ..., a_n]$. Notice that the length of a is not n because some words are frozen. Finally, we choose the word which has the max important score as the vulnerable word:

$$m = argmax\,(a), \tag{10}$$

where m is the index of the vulnerable word and $argmax$ is the function to pop the word which has the most important score and get its index.

After finding the vulnerable word, we proceed to locate all synonyms of w_m in order to substitute it:

$$S = Synonyms\,(w_m), \tag{11}$$

where S is the synonyms set of w_m, we sequentially select a word in S based on the similarity to w_m then substitute w_m:

$$s' = MaxSim\,(S, w_m), \tag{12}$$

$$x^s = [w_1, w_2, , w_{m-1}, s', w_{m+1}, ..., w_n], \tag{13}$$

where x^s is the sentence by replacing w_m in x with s'. $MaxSim$ is the function to pop the word in S that is most similar to w_m. Notice that if S is already empty before popping, we go back to Eqn. (10) and repeat the above process. After obtaining x^s, we perform different actions based on the following situations, if $f\,(x^s) \neq f\,(x)$, the final adversarial sample x^* is x^s:

$$x^* = x^s. \tag{14}$$

If $f\,(x^s) = f\,(x)$ and $prob\,(f\,(x^s)) < prob\,(f\,(x))$, we will keep this word change:

$$x = x^s. \tag{15}$$

Then go back to Eqn. (12) and repeat the above process. If $f\,(x^s) = f\,(x)$ and $prob\,(f\,(x^s)) \geq prob\,(f\,(x))$, we will abandon this word change then go back to Eqn. (12) and repeat the above process.

In our attacker, we utilize BertAttack (Li et al. 2020) as our backbone, which utilizes *[mask]* token to mask words and bert embedding to calculate the similarity of words.

Experiments

Experimental Setting

Victim Models We choose four mainstream large language models as our victim models.

- **Flan-T5-large** (Chung et al. 2022): Flan-T5-large is a derivative of the Text-to-Text Transfer Transformer (T5) model, developed by Google. It has 760M parameters.
- **Flan-T5-xl** (Chung et al. 2022): Flan-T5-xl is a large version of Flan-T5 than Flan-T5-large, developed by Google. It has 3B parameters.
- **ChatGLM2** (Du et al. 2022): ChatGLM2 is the second-generation version of the open-source bilingual (Chinese-English) chat model ChatGLM, developed by Tsinghua University. It has 6B parameters.
- **ChatGPT** (OpenAI 2023): Developed by OpenAI, ChatGPT is a large language model trained to generate human-like text. It uses the GPT-3 architecture and has been fine-tuned for more interactive and conversational tasks. In detail, we use the gpt-3.5-turbo API.

We set the temperature = 0 to stabilize the output of LLMs. When attacking victim models, we not only attack them with zero-shot prompt but also few-shot prompt. Specifically, we employ four MWP samples as shots and provide Chain-of-Thought (CoT) (Wei et al. 2022) annotations. This few-shot prompt serves as a method to enhance the math solving ability of LLMs. Similar with other prompts, they are not changed during the attack process.

Datasets Two math word problems benchmark datasets **GSM8K** (Cobbe et al. 2021) and **MultiArith** (Roy and Roth 2015) are adopted in the experiments. However, we only select the subsets for the following considerations by following the previous work (Zhu et al. 2023): (1) we focus on simple MWPs, because hard samples have very lower accuracy not necessary to attack. (2) Owing to the extensive computational requirements of generating single adversarial sample, which necessitates iterating over the entire dataset 100 times on average. Finally, for GSM8K, we firstly remove those hard samples labelled by more than three solving steps, then randomly select half of those remained simple MWPs and obtain 307 MWP samples. For MultiAirth, all MWPs are simple thus we randomly select 150 MWPs similar with the previous work (Zhu et al. 2023).

Metrics Given a dataset D with N data instance x and label y, victim model f, an adversarial attack method A that generates adversarial examples $A(x)$, we adopt following four metrics:

- **Clean Acc**: The accuracy before attacking. Clean Acc = $\frac{\sum_{(x,y)\in D} \mathbb{I}[f(x)=y]}{N}$.
- **Attack Acc**: The accuracy after attacking. Attack Acc = $\frac{\sum_{(x,y)\in D} \mathbb{I}[f(x)=y \cap f(A(x))=y]}{N}$.
- **Attack Success Rate (ASR)** (Wang et al. 2023a): The rate of samples is successfully attacked. ASR = $\frac{\sum_{(x,y)\in D} \mathbb{I}[f(A(x))\neq y]}{\sum_{(x,y)\in D} \mathbb{I}[f(x)=y]}$.
- **Similarity**: The average semantic similarity between the adversarial sample and the original sample. We use Universal Sentence Encoder (Cer et al. 2018) to measure semantic similarity.

To ensure the correctness in the experiments, we check each adversarial sample manually, and consider adversarial examples which are changed mathematical logic as unsuccessful attacks.

Main Results

As shown in Table 1, our approach can effectively attack the math solving ability of large language models. For

		GSM8K				MultiAirth			
Prompt	**Models**	**Clean Acc**	**Attack Acc**	**ASR**	**Similarity**	**Clean Acc**	**Attack Acc**	**ASR**	**Similarity**
Zero shot	Flan-T5-large	18.24	2.28	87.50	90.55	2.00	0.00	100.00	91.42
	Flan-T5-xl	21.17	3.58	83.08	92.86	7.33	0.67	90.90	95.63
	ChatGLM2	54.40	23.78	56.29	91.58	71.33	20.67	71.03	94.00
	ChatGPT	84.69	49.54	41.15	89.26	98.67	60.00	39.19	91.33
Few shot	Flan-T5-large	22.15	10.42	52.94	92.92	5.33	0.67	87.5	95.66
	Flan-T5-xl	32.35	17.59	45.45	90.00	10.67	2.67	75.00	95.16
	ChatGLM2	64.82	22.80	64.82	90.71	37.33	7.33	80.36	95.75
	ChatGPT	88.27	70.68	19.93	87.19	98.00	77.33	21.09	86.97

Table 1: Results of attacking against various large language models.

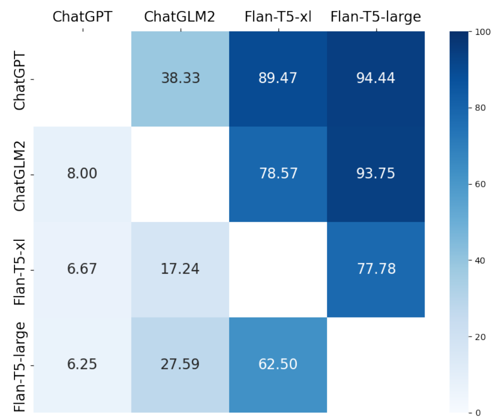

Figure 3: Transfer Success Rate (TSR) of Y-axis models to X-axis models. The generated adversarial samples of larger-size models can attack smaller-size models.

LLMs with zero-shot, we could get the high ASR, even for ChatGPT, it could achieve an average of 40% on GSM8K (41.15%) and MultiAirth (39.19%). The average Similarity is large than 90%, indicating that we can successfully generate adversarial samples with high similarity and do not alter mathematical logic.

Comparing different LLMs, we can observe that more powerful LLMs (i.e., higher Clean Acc) are more difficult to attack (i.e., lower ASR). For Flan-T5-large and Flan-T5-xl, their robustness in math solving ability is poor, as even a slight disturbance can cause them to predict incorrectly. For ChatGLM2 and ChatGPT, their robustness is noticeably stronger, as our method fails to attack them on some MWP samples.

Furthermore, comparing zero-shot and few-shot, we can see that employing few-shot could enhance the math solving ability of the LLMs and also make them more robust, leading to a lower ASR. For models with stronger in-context ability, the enhancement becomes larger. Like ChatGPT, the Attack Success Rate could decrease from 41.15% to 19.93%. However, we find ChatGLM2 exhibits poor in-context ability which leads math solving ability as well as robustness does not improve with the few-shot prompt.

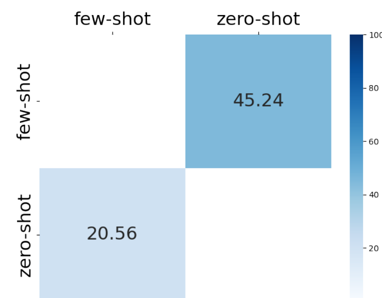

Figure 4: Transfer Success Rate (TSR) of Y-axis prompt to X-axis prompt. The generated adversarial samples of model with few-shot can attack model with zero-shot.

Fine-grained Analysis

Transferability To test the transferability of the generated adversarial samples, we take adversarial samples of model A to attack other models B. Specifically, we select the samples that B can correctly predict as the experimental samples. Subsequently, we provide B with adversarial samples generated by attacking A on experimental samples. We examine if these adversarial samples can successfully attack model B. Here, we propose the metric: Transfer Success Rate (**TSR**), if an adversarial sample of A can successfully attack model B then it means transfer success.

In Figure 3, we show the TSR between Y-axis (i.e., A model) and X-axis (i.e., B model) models, and we can observe that the adversarial samples of larger-size models can attack smaller-size models too. ChatGPT could get 94.44% TSR to Flan-T5-large and 89.47% TSR to Flan-T5-xl. Specifically, we find that the TSR will increase when the math solving ability between models grows wider. As shown in Figure 3, we can see that the adversarial samples of smaller-size models can not attack larger-size models. Flan-T5-large and Flan-T5-xl both show low TSR (6.25% and 6.67%) on ChatGPT. In this experiment, all tested models are in zero-shot setting.

In order to see the transferability performance between zero-shot and few-shot, we conducted the same experiment on ChatGPT. As shown in Figure 4, the ChatGPT with few-shot can achieve 45.24% TSR to ChatGPT with zero-shot however the reverse is only 20.56%. It indicates that the ad-

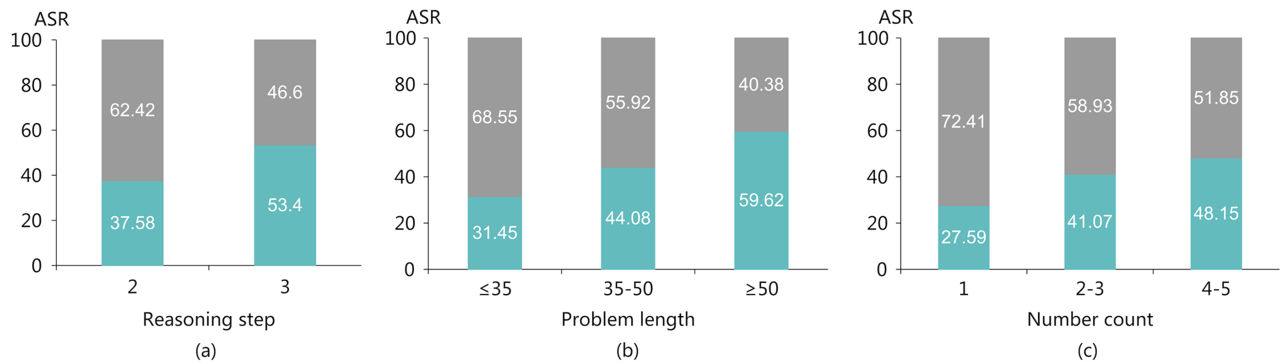

Figure 5: Analysis which MWPs are easier to attack. (a) shows the effect of answer reasoning steps on the ASR. (b) shows the effect of problem length on the ASR. (c) shows the effect of numbers' count in MWP on the ASR.

	Clean Acc	Attack Acc	ASR	Similarity
GSM8K	87.95	75.57	14.07	88.33
MultiAirth	98.00	82.00	16.33	88.15

Table 2: Results of attacking against large language models with adversarial samples prompt on ChatGPT.

versarial samples of LLM with few-shot can attack that with zero-shot. And adversarial samples of LLM with zero-shot can not transfer to that with few-shot.

Analysis on MWPs To know which MWPs are easier to attack, we investigate the effects of MWPs reasoning steps, problem length and number count on ASR. Specifically, we conducted the experiment on ChatGPT in zero-shot setting. As shown in Figure 5: (a) with the increase of reasoning steps of ground truth, we can observe that the ASR will increase when the reasoning steps from 2 to 3. Reasoning steps of ground truth can be regarded as a metric to measure the difficulty of MWP. Difficult MWPs are easier to attack. (b) with the increase in problem length, we can observe a gradual increase in the ASR as the length of the math word problems become longer. Long MWPs are easier to attack. (c) with the increase in the quantity of numbers in MWPs, we can observe a gradual increase in ASR as the number counts become more. All the above three factors can be used to measure the complexity of an MWP (Fu et al. 2023b), therefore, we can draw a conclusion that more complex MWPs are easier to attack. It shows that LLMs are sensitive to disturbances on MWPs with complex mathematical logic.

Using Attacking Samples as Prompts We also study the impact of adversarial samples on improving the robustness of large language models. In the few-shot prompts, we replace normal MWP examples by corresponding adversarial examples generated by our MathAttack but with correct labels, and observe their impact on the math solving ability and robustness of the LLMs. As shown in Table 2, we can see the Clean ACC still maintains a high level of accuracy (87.95% on GSM8K and 98.00% on MultiAirth), because the adversarial examples generated by our MathAttack ex-

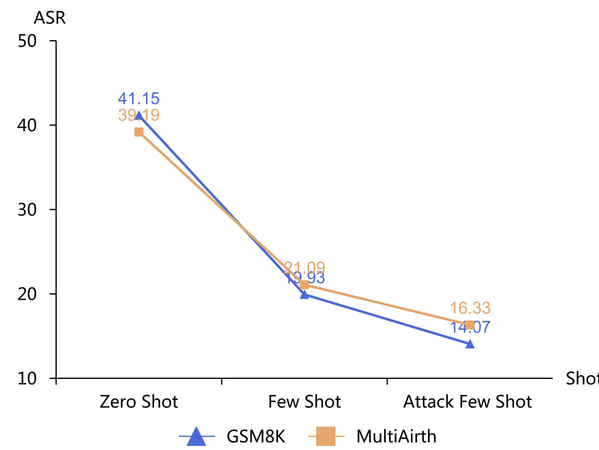

Figure 6: Trend of ASR after utilizing different prompts, Attack Few Shot means replacing the MWP samples of Few Shot to their corresponding adversarial samples.

hibit high similarity to the original samples. By comparing the Attack Acc and ASR in Table 1, it is surprised to find the use of adversarial examples in the few-shot prompts can enhance the robustness of LLMs (i.e., much lower ASR by comparing to the normal results in Table 1). When we use adversarial examples in the prompt, the LLM could see these examples that are disturbed but still able to predict correctly, therefore they will not affected by some small disturbances when predict. Figure 6 provides a more intuitive visualization, demonstrating that the robustness of LLMs utilizing few-shot prompt can be significantly improved by comparing to zero-shot prompt. When employing adversarial examples as few-shot prompt, it will further strengthen their robustness and the ASR of large language models further decrease. This observation motivates us to enhance the robustness of large language models without compromising their math solving ability by employing the adversarial examples generated by MathAttack as few-shot prompt.

Orginal Sample	Adversarial Sample
A **class** of 50 students has various hobbies. 10 like to bake 5 like to play basketball and the rest like to either play video games or play music. How many like to play video games if the number that like to play music is twice the number that prefer playing basketball?	A **group** of 50 students has various hobbies. 10 like to bake 5 like to play basketball and the rest like to either play video games or play music. How many like to play video games if the number that like to play music is twice the number that prefer playing basketball?
Solution: Let's assume the number of students who like to play video games is "x". **......** So, we can set up an equation: 10 + 5 + x + 10 = 50. Simplifying the equation: 10 + 5 + x + 10 = 50. 25 + x = 50. x = 50 - 25. x = 25. Therefore, the number of students who like to play video games is 25. (**Correct!**)	**Solution**: Let's assume the number of students who like to play video games is "x". **......** We can set up the equation: x + 10 = 50. Now, let's solve for x: x = 50 - 10. x = 40. Therefore, the number of students who like to play video games is 40. (**Wrong!**)

Table 3: A real case predicted by ChatGPT on original MWP (left) and its adversarial sample (right).

	Zero-shot		Few-shot	
	Original	RobustMath	Original	RobustMath
Flan-T5-large	10.75	4.67	13.08	10.67
Flan-T5-xl	17.76	6.00	26.17	20.33
Flan-T5-xl-F	16.36	12.33	10.19	9.33
ChatGLM2	47.08	36.67	54.67	33.67

Table 4: Accuracy of large language models on RobustMath and its original samples set. Flan-T5-xl-F is the finetuned model on 200k MWP data (Fu et al. 2023a).

Case Study

Table 3 reports a real case predicted by ChatGPT on the original MWP and its adversarial sample generated by MathAttack. We can find that the adversarial sample generated by MathAttack is similar to the original sample with few changes. For the original sample, ChatGPT can give the correct reasoning process step by step and finally get the correct answer *25*. But when MathAttack simply changed *class* in the original sample to *group*, ChatGPT can come up with the wrong reasoning process and get the wrong equation (x+*10=50*), end up with the wrong answer *40*. These cases show that the robustness of LLMs in math solving ability still needs to be strengthened.

New MWP Dataset RobustMath

Using the transferability of adversarial samples, we attack ChatGPT by MathAttack to build our **RobustMath** dataset. Specifically, we first utilize GSM8K and MultiAirth as our seed data then attack ChatGPT to generate adversarial samples. After that, in order to ensure the high quality of RobustMath, we manually check each adversarial sample and filter out samples which change the mathematical logic. Ultimately, our RobustMath has 300 high-quality adversarial samples that can be used to measure the robustness of large language models' math solving ability.

To verify the effectiveness of our RobustMath, we evaluate large language models on RobustMath. In addition to the models mentioned above, we also evaluate large language model that is fine-tuned on MWP datasets. Specifically, we follow (Fu et al. 2023a) to finetune Flan-T5-xl with 200k MWP data. In Table 4, we observe that the performance of the LLMs on RobustMath is significantly worse compared to the performance on its original samples. This indicates that our RobustMath can effectively measure the robustness of the model's math solving ability. When examining the performance of models with zero-shot performance, we can see that as the model's capability increases, its performance on RobustMath also increases. However, it still does not exceed 37.00%. Moreover, after finetuning, the performance of Flan-T5-xl increases from 6.00% to 12.33%. It indicates that finetuning on specific data could help improve the robustness of models. Observing the performance of models with few-shot prompt, we find that models with a strong in-context ability such as Flan-T5-large and Flan-T5-xl can effectively enhance their performance on RobustMath. In contrast, ChatGLM2 and finetuned Flan-T5-xl which have weaker in-context ability do not exhibit significant improvements on both the original samples set and RobustMath under few-shot prompt.

Conclusion and Future Work

In this paper, we make a first attempt to attack MWP samples to examine the security of LLMs in math solving ability. To preserve the mathematical logic of MWPs, we propose a **MathAttack** model with a logical entity recognition block. Extensive experiments show that MathAttack could effectively attack the math solving ability. Through the comprehensive experimental analysis, we have three significant findings: (1) Transferability of attacking samples (2) Complex MWPs (such as more solving steps, longer text, more numbers) are more vulnerable to attack, and (3) Attacking samples used in few-shot prompts can improve robustness of LLMs. Furthermore, we propose a new dataset **RobustMath** by utilizing MathAttack and manual check, which consists of high-quality MWP Adversarial samples and could measure the robustness of LLMs' math solving ability. We hope our practice and observations can serve as an important attempt to enhance the robustness of LLMs in math solving ability. In the future, we will explore methods such as instruction learning or reinforcement learning to enhance the robustness of the models. As large language models are increasingly being applied in the field of intelligence education, the importance of improving their robustness becomes more significant.

Acknowledgements

The work was partially supported by the following: National Natural Science Foundation of China under No. 92370119, No. 62376113, and No. 62276258; Jiangsu Science and Technology Programme (Natural Science Foundation of Jiangsu Province) under no. BE2020006-4, European Union's Horizon 2020 research and innovation programme no. 956123, and UK EPSRC under projects [EP/T026995/1].

References

Cer, D.; Yang, Y.; Kong, S.-y.; Hua, N.; Limtiaco, N.; John, R. S.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Zhu, K.; Chen, H.; Yang, L.; Yi, X.; Wang, C.; Wang, Y.; et al. 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*.

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

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, E.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Du, Z.; Qian, Y.; Liu, X.; Ding, M.; Qiu, J.; Yang, Z.; and Tang, J. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Proceedings of the 2022 conference on Annual Meeting of the Association for Computational Linguistics*, 320–335.

Fu, Y.; Peng, H.; Ou, L.; Sabharwal, A.; and Khot, T. 2023a. Specializing Smaller Language Models towards Multi-Step Reasoning.

Fu, Y.; Peng, H.; Sabharwal, A.; Clark, P.; and Khot, T. 2023b. Complexity-based prompting for multi-step reasoning.

Handa, K.; Clapper, M.; Boyle, J.; Wang, R. E.; Yang, D.; Yeager, D.; and Demszky, D. 2023. Mistakes Help Us Grow": Facilitating and Evaluating Growth Mindset Supportive Language in Classrooms. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Ji, H.; Han, I.; and Ko, Y. 2023. A systematic review of conversational AI in language education: Focusing on the collaboration with human teachers. *Journal of Research on Technology in Education*, 55(1): 48–63.

Jin, D.; Jin, Z.; Zhou, J. T.; and Szolovits, P. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the 2020 conference on Association for the Advancement of Artificial Intelligence*, volume 34, 8018–8025.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213.

Kumar, V.; Maheshwary, R.; and Pudi, V. 2021. Adversarial Examples for Evaluating Math Word Problem Solvers. In *Findings of the Association for Computational Linguistics: 2021 conference on Empirical Methods in Natural Language Processing*, 2705–2712.

Kumar, V.; Maheshwary, R.; and Pudi, V. 2022. Practice makes a solver perfect: Data augmentation for math word problem solvers. 4194–4206.

Li, A.; Xiao, Y.; Liang, J.; and Chen, Y. 2022. Semantic-based data augmentation for math word problems. In *Proceedings of the 2022 conference on Database Systems for Advanced Applications*, 36–51. Springer.

Li, J.; Ji, S.; Du, T.; Li, B.; and Wang, T. 2019. Textbugger: Generating adversarial text against real-world applications.

Li, L.; Ma, R.; Guo, Q.; Xue, X.; and Qiu, X. 2020. Bert-attack: Adversarial attack against bert using bert. 6193–6202.

Macina, J.; Daheim, N.; Chowdhury, S. P.; Sinha, T.; Kapur, M.; Gurevych, I.; and Sachan, M. 2023. MathDial: A Dialogue Tutoring Dataset with Rich Pedagogical Properties Grounded in Math Reasoning Problems. *arXiv preprint arXiv:2305.14536*.

OpenAI. 2023. Gpt-4 technical report.

Patel, A.; Bhattamishra, S.; and Goyal, N. 2021. Are NLP Models really able to Solve Simple Math Word Problems? In *Proceedings of the 2021 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2080–2094.

Qian, Z.; Huang, K.; Wang, Q.-F.; and Zhang, X.-Y. 2022. A survey of robust adversarial training in pattern recognition: Fundamental, theory, and methodologies. *Pattern Recognition*, 131: 108889.

Qiao, S.; Ou, Y.; Zhang, N.; Chen, X.; Yao, Y.; Deng, S.; Tan, C.; Huang, F.; and Chen, H. 2022. Reasoning with language model prompting: A survey. *arXiv preprint arXiv:2212.09597*.

Roy, S.; and Roth, D. 2015. Solving General Arithmetic Word Problems. In *Proceedings of the 2015 conference on Empirical Methods in Natural Language Processing*, 1743–1752.

Shi, F.; Chen, X.; Misra, K.; Scales, N.; Dohan, D.; Chi, E. H.; Schärli, N.; and Zhou, D. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 2023 conference on International Conference on Machine Learning*, 31210–31227. PMLR.

Uesato, J.; Kushman, N.; Kumar, R.; Song, F.; Siegel, N.; Wang, L.; Creswell, A.; Irving, G.; and Higgins, I. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

Wang, J.; Liu, Z.; Park, K. H.; Chen, M.; and Xiao, C. 2023a. Adversarial Demonstration Attacks on Large Language Models. *arXiv preprint arXiv:2305.14950*.

Wang, J.; Xixu, H.; Hou, W.; Chen, H.; Zheng, R.; Wang, Y.; Yang, L.; Ye, W.; Huang, H.; Geng, X.; et al. 2023b. On the Robustness of ChatGPT: An Adversarial and Out-of-distribution Perspective. In *International Conference on Learning Representations 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*.

Wang, L.; Xu, W.; Lan, Y.; Hu, Z.; Lan, Y.; Lee, R. K.-W.; and Lim, E.-P. 2023c. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

Wang, R. E.; and Demszky, D. 2023. Is ChatGPT a Good Teacher Coach? Measuring Zero-Shot Performance For Scoring and Providing Actionable Insights on Classroom Instruction. *arXiv preprint arXiv:2306.03090*.

Wang, R. E.; Zhang, Q.; Robinson, C.; Loeb, S.; and Demszky, D. 2023d. Step-by-Step Remediation of Students' Mathematical Mistakes. *arXiv preprint arXiv:2310.10648*.

Wang, Y.; Liu, X.; and Shi, S. 2017. Deep neural solver for math word problems. In *Proceedings of the 2017 conference on Empirical Methods in Natural Language Processing*, 845–854.

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

Xie, Z.; and Sun, S. 2019. A Goal-Driven Tree-Structured Neural Model for Math Word Problems. In *Proceedings of the 2019 conference on International Joint Conference on Artificial Intelligence*, 5299–5305.

Xu, H.; Ma, Y.; Liu, H.-C.; Deb, D.; Liu, H.; Tang, J.-L.; and Jain, A. K. 2020. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17: 151–178.

Yao, J.; Zhou, Z.; and Wang, Q. 2023. Solving Math Word Problem with Problem Type Classification. In *CCF International Conference on Natural Language Processing and Chinese Computing*, 123–134. Springer.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T. L.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*.

Ye, M.; Miao, C.; Wang, T.; and Ma, F. 2022. TextHoaxer: budgeted hard-label adversarial attacks on text. In *Proceedings of the 2022 conference on Association for the Advancement of Artificial Intelligence*, volume 36, 3877–3884.

Zhang, J.; Wang, L.; Lee, R. K.-W.; Bin, Y.; Wang, Y.; Shao, J.; and Lim, E.-P. 2020. Graph-to-tree learning for solving math word problems. In *Proceedings of the 2020 conference on Annual Meeting of the Association for Computational Linguistics*, 3928–3937.

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; et al. 2023a. Least-to-most prompting enables complex reasoning in large language models.

Zhou, Z.; Ning, M.; Wang, Q.; Yao, J.; Wang, W.; Huang, X.; and Huang, K. 2023b. Learning by Analogy: Diverse Questions Generation in Math Word Problem. In *Findings of the Association for Computational Linguistics*, 11091–11104.

Zhu, K.; Wang, J.; Zhou, J.; Wang, Z.; Chen, H.; Wang, Y.; Yang, L.; Ye, W.; Gong, N. Z.; Zhang, Y.; et al. 2023. PromptBench: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts. *arXiv preprint arXiv:2306.04528*.

Zhuo, T. Y.; Li, Z.; Huang, Y.; Li, Y.-F.; Wang, W.; Haffari, G.; and Shiri, F. 2023. On robustness of prompt-based semantic parsing with large pre-trained language model: An empirical study on codex. *arXiv preprint arXiv:2301.12868*.