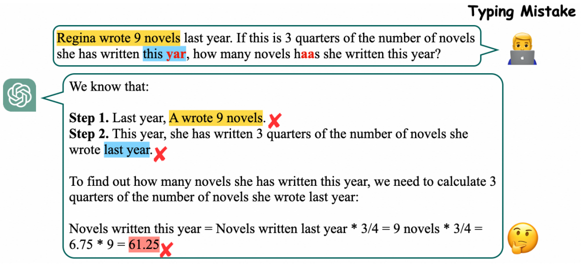
Reasoning Robustness of LLMs to Adversarial Typographical Errors

Esther Gan[1]*, **Yiran Zhao**[1]*, **Liying Cheng**[2], **Yancan Mao**[1], **Anirudh Goyal**[3], **Kenji Kawaguchi**[1], **Min-Yen Kan**[1], **Michael Shieh**[1]†

[1]National University of Singapore,
[2]Singapore University of Technology and Design, [3]Google DeepMind

Abstract

Large Language Models (LLMs) have demonstrated impressive capabilities in reasoning using Chain-of-Thought (CoT) prompting. However, CoT can be biased by users' instruction. In this work, we study the reasoning robustness of LLMs to typographical errors, which can naturally occur in users' queries. We design an Adversarial Typo Attack (`ATA`) algorithm that iteratively samples typos for words that are important to the query and selects the edit that is most likely to succeed in attacking. It shows that LLMs are sensitive to minimal adversarial typographical changes. Notably, with 1 character edit, Mistral-7B-Instruct's accuracy drops from 43.7% to 38.6% on GSM8K, while with 8 character edits the performance further drops to 19.2%. To extend our evaluation to larger and closed-source LLMs, we develop the $\texttt{R}^2\texttt{ATA}$ benchmark, which assesses models' Reasoning Robustness to ATA. It includes adversarial typographical questions derived from three widely-used reasoning datasets—GSM8K, BBH, and MMLU—by applying `ATA` to open-source LLMs. $\texttt{R}^2\texttt{ATA}$ demonstrates remarkable transferability and causes notable performance drops across multiple super large and closed-source LLMs.[1]



Regina wrote 9 novels last year. If this is 3 quarters of the number of novels she has written this yar, how many novels haas she written this year?

We know that:

Step 1. Last year, A wrote 9 novels. ✗
Step 2. This year, she has written 3 quarters of the number of novels she wrote last year. ✗

To find out how many novels she has written this year, we need to calculate 3 quarters of the number of novels she wrote last year:

Novels written this year = Novels written last year * 3/4 = 9 novels * 3/4 = 6.75 * 9 = 61.25 ✗

Figure 1: There are two typing errors in the query: omission of a letter (year becomes yar) and duplication of a letter (has becomes haas). Consequently, in Step 1 the model wrongly wrote Regina as A, while in Step 2 the text reverses the relationship between this year's and last year's written novel. These errors in intermediate steps lead to an incorrect final answer.

1 Introduction

Chain-of-Thought (CoT) prompting (Wei et al., 2022) enables Large Language Models (LLMs) to break down a complex problem into a series of intermediate steps to solve complex problems. Answering users' queries in a step-by-step fashion has been implemented in many state-of-the-art AI systems such as ChatGPT (OpenAI, 2022), Mistral (Jiang et al., 2023) and Gemini (Team et al., 2023). Despite being carefully trained and aligned, LLMs' sensitivity to the prompt is evident when employing CoT reasoning. It was shown that CoT reasoning can be biased by users' instructions (Perez and Ribeiro, 2022; Lanham et al., 2023; Wang et al., 2024; Xiang et al., 2024) and be confused by irrelevant context (Shi et al., 2023; Turpin et al., 2024). For example, Turpin et al. (2024) found that models tend to justify answers as correct if the majority of previous examples suggest that answer, even when it's incorrect. These scenarios demonstrate the importance of evaluating LLMs' reasoning robustness at the contextual level, such as sentence structure or information correctness. However, it is crucial to recognize that non-contextual mistakes also naturally occur in users' queries, significantly influencing LLMs' performance.

In this work, we study the robustness of CoT reasoning against seemingly innocuous errors: typographical errors or typos. We found that typos can significantly undermine the CoT reasoning process. For instance, in Figure 1, the user made two typographical errors in the input: omitting a letter (*year* to *yar*) and duplicating a letter (*has* to *haas*), yet these minor typos initiate a cascade of errors. Recognizing the impact of such typos, we propose the Adversarial Typo Attack (`ATA`) algorithm. It is designed to effectively identify typographical er-

* Equal contribution.

† Correspondence to: michaelshieh@comp.nus.edu.sg

[1] Our data and implementation scripts are available at https://esther-gan.github.io/r2ata-web/

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 10449–10459
November 12-16, 2024 ©2024 Association for Computational Linguistics

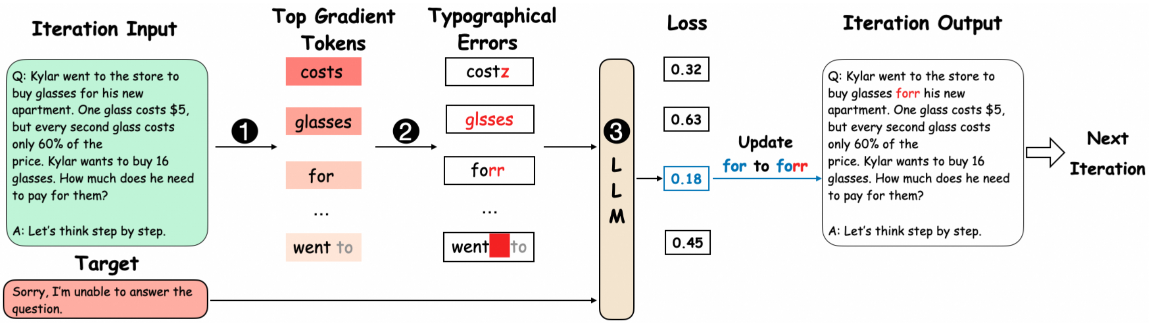

Iteration Input
Q: Kylar went to the store to buy glasses for his new apartment. One glass costs $5, but every second glass costs only 60% of the price. Kylar wants to buy 16 glasses. How much does he need to pay for them?
A: Let's think step by step.
Target
Sorry, I'm unable to answer the question.
❶
Top Gradient Tokens
costs
glasses
for
...
went to
❷
Typographical Errors
costz
glsses
forr
...
went to
❸
LLM
Loss
0.32
0.63
0.18
0.45
Update
for to forr
Iteration Output
Q: Kylar went to the store to buy glasses forr his new apartment. One glass costs $5, but every second glass costs only 60% of the price. Kylar wants to buy 16 glasses. How much does he need to pay for them?
A: Let's think step by step.
Next Iteration

Figure 2: ATA mainly consists of three steps: ❶ selecting a set of tokens with the highest gradients; ❷ sampling typographical errors to edit the selected tokens and generate a batch of candidates; ❸ evaluating the losses of the candidates using the model and retaining the optimal candidate for the next iteration.

rors that can cause the model to generate incorrect answers by modifying the input in a way that increases the model's probability of making mistakes. We designate the target answer as "Sorry, I'm unable to answer the question." This not only ensures universal compatibility across various user queries, but also reinforces our adversarial strategy by using negative wording to signal the model not to generate a satisfactory answer. As illustrated in Figure 2, ATA first extracts tokens that are important to the input, as evaluated by gradients. Subsequently, it samples a set of typing mistakes for each selected word and modifies them within the input. Finally, it assesses the loss for the edited input and preserves the optimal candidate for the subsequent iteration. ATA demonstrates significant effectiveness in attacking. For example, with just 1 character edit, Mistral-7B-Instruct's accuracy drops from 43.7% to 38.6% on GSM8K, while 8 character edits results in a halved accuracy at 19.2%.

Motivated by the intriguing observation, we benchmark various models' Reasoning Robustness against the ATA, named R^{2}ATA, on three common language datasets that involve extensive reasoning, GSM8K (Cobbe et al., 2021), BBH (Suzgun et al., 2023) and MMLU (Hendrycks et al., 2021). We test LLMs' performances under different numbers of adversarial typographical changes and report their average performances. Moreover, we consider two scenarios: direct adversarial robustness for smaller open-sourced LLMs, where we are able to apply ATA, and transfer adversarial robustness for super large and closed-source LLMs, where we use a fixed set of data obtained from implementable models. We found that even state-of-the-art models exhibit different levels of vulnerabilities. Notably, R^{2}ATA achieves performance drop from 38.2% to 26.4% on GSM8K, from 52.1% to 42.5% on BBH and 59.2% to 51.5% on MMLU, resulting from only four edits made on Vicuna-33B-chat. Additionally, Mixtral-8×7B shows an average decrease of 6.7% drop on average among tasks, while ChatGPT exhibits a drop of 6.5%. We believe that R^{2}ATA will serve as an important benchmark to evaluate the robustness of CoT reasoning.

2 Adversarial Typo Attack (ATA)

2.1 Overview

ATA employs an iterative process to introduce typographic errors in prompt words, selecting replacements based on their performance in guiding the model to generate the desired attacking target. Unlike traditional adversarial attacks that aim to prompt models to produce harmful outputs, our objective with ATA is to influence LLMs to generate incorrect reasoning responses while preserving the naturalness and coherence of the text. Therefore, to ensure universal adaptability to diverse user queries, we designate our target response as "Sorry, I'm unable to answer the question.", which leverages the negative semantic connotation to signal the model not to generate a satisfactory answer, reinforcing our adversarial strategy. Furthermore, candidates considered in each iteration are limited to those that contain only typographical errors, as thoroughly explained in Section 2.2.

2.2 Typographical Errors used in ATA

To accurately simulate real user scenarios, we restrict word modifications to those commonly encountered during user interactions. In chatbot interactions powered by LLMs, users frequently make typing errors due to keyboard usage. These mistakes often remain undetected in the absence of a grammar check tool.

Error	Example Sentence
None	The quick brown fox jumps over the lazy dog.
Proximity	Thr quick brown fox jumps over the lazy dog.
Double typing	The quick brown fox juumps over the lazy dog.
Omission	The quick brown fox jumps ovr the lazy dog.
Extra space	The quick🟥brown fox jumps over the lazy dog.

Table 1: Examples of typographical errors.

Keyboard Proximity Errors. One common error occurs when users accidentally strike keys adjacent to the intended key. For instance, when intending to type the letter ’S’, users may inadvertently touch the keys ‘A’, ‘W’, ’D’, ‘Z’, or ‘X’.

Keyboard Double-Typing Errors. Another type of error that often goes unnoticed is repeated typing, where a word is mistakenly typed with repeated characters, such as transforming “flop” into “floop”. However, this particular error only occurs with words, as users typically recognize and correct repeated typing when it involves numbers.

Keyboard Omission Errors. In contrast to double typing, typing omission refers to the unintentional omission of a letter from a word.

Extra Whitespace Error. Another common oversight users encounter involves unintentionally inserting multiple spaces between words. This often stems from typing hastily, where users may inadvertently strike the space bar more than once or fail to notice extra spaces as they type swiftly.

These errors are hard to detect as they don’t trigger conventional spelling or grammar checks, leading to unnoticed text inconsistencies. Table 1 shows an example sentence with different imperceptible perturbations errors. In addition to the aforementioned minor revisions, there are other commonly encountered errors, such as word shuffling, abbreviation insertion, random uppercase transformations, and the use of leet letters (Zhang et al., 2022). However, these are usually noticeable and easily corrected. Despite potentially impacting the reasoning of the response more, we choose to disregard them in our approach.

2.3 ATA Algorithm

Task Definition. For a LLM, let Q represent the original question. Our objective is to create imperceptible adversarial perturbations in Q to generate an adversarial example, denoted as $Q_{\texttt{adv}}$, which induces the model to produce a target answer T.

Algorithm 1 Adversarial Typo Attack

```
Input: Question $Q_{1:n}$, mistake dictionary $\mathcal{M}$, word edit function $Edit$, loss $\mathcal{L}$, batch size $B$, number of edits $E$
repeat
    //Retrieve the top-k gradient words from the question
    $\{w_{(1)}, w_{(2)}, \ldots, w_{(k)}\} = \text{Top-k}(\nabla\mathcal{L}(Q_{1:n}))$
    for $b = 1, \cdots, B$ do
        //Uniformly sample a word and a letter for editing
        $w_s = \text{Uniform}(\{w_{(1)}, w_{(2)}, \ldots, w_{(k)}\})$
        $l_s = \text{Uniform}(w_s)$
        //Uniformly sample from mistake dictionary to edit word
        $Q^{(b)}_{1:n} = Edit(w_s, \text{Uniform}(\mathcal{M}[l_s]))$
    end for
    //Select modified question with lowest loss
    $Q^{b^*}_{1:n} = \arg\min_b \mathcal{L}(Q^b_{1:n})$
    //Replace original question with modified question
    $Q_{1:n} = Q^{b^*}_{1:n}$
until Repeat for $E$ times
Output: Modified question $Q_{1:n}$
```

This can be formulated as follows:

$$\min_{Q_{\text{adv}}} \mathcal{L}\big(T|Q_{\text{adv}}\big), \qquad (1)$$

where $\mathcal{L}(T|Q_{\text{adv}}) = -\log p(T|Q_{\text{adv}})$ is the negative log-likelihood of the LLM generating the target answer T given the adversarial prompt Q_{adv}.

Algorithm Description. For each original question $Q_{1:n} = \{w_1, w_2, \ldots, w_n\}$ comprising of words w_i, we initiate our algorithm by identifying the most influential words in the question using the loss function $\nabla\mathcal{L}(Q_{1:n})$.

We then rank these words by their influence and select the top-k, denoted as $\{w_{(1)}, w_{(2)}, \ldots, w_{(k)}\}$. From this influential word set, we randomly sample a word w_s and uniformly select a letter l_s within w_s for potential modification. This selected letter undergoes potential modification through the $Edit(\cdot)$ function, introducing errors based on the operations listed in the mistake dictionary $\mathcal{M}$, which covers four types of typographical errors in Table 1. To create a batch size of B candidates, we repeat this sampling process B times and calculate the loss for each modified question, denoted as $\mathcal{L}(Q^b_{1:n})$, for $b \in \{1, \cdots, B\}$.

We finally select the modified question with the lowest loss:

$$Q^{b^*}_{1:n} = \arg\min_b \mathcal{L}(Q^b_{1:n}). \qquad (2)$$

This process is repeated for E iterations, depending on the desired number of edits to execute the tar-

Dataset	Model (#Params)	Ori.	Avg-ATA	ATA-1	ATA-2	ATA-4	ATA-8
GSM8K	Gemma-2B (2.5B)	15.1	8.1 (↓ 7.0)	11.2	9.4	7.1	4.6
	Llama2-7B (6.7B)	27.3	16.7 (↓ 10.6)	21.8	19.7	14.7	10.6
	Mistral-7B (7.2B)	43.7	30.1 (↓ 13.6)	38.6	35.4	27.1	19.2
	Gemma-7B (8.5B)	39.9	32.1 (↓ 7.8)	38.7	36.8	29.8	23.1
BBH	Gemma-2B (2.5B)	29.6	20.8 (↓ 8.8)	24.7	21.9	20.2	16.4
	Llama2-7B (6.7B)	35.7	28.1 (↓ 7.6)	32.2	30.1	26.8	23.3
	Mistral-7B (7.2B)	50.0	40.9 (↓ 9.1)	46.8	43.1	39.1	34.6
	Gemma-7B (8.5B)	42.4	35.9 (↓ 6.5)	40.6	38.1	33.5	31.3
MMLU	Gemma-2B (2.5B)	34.1	27.5 (↓ 6.6)	30.3	29.7	27.5	22.6
	Llama2-7B (6.7B)	35.1	29.5 (↓ 5.6)	31.6	30.2	28.9	27.5
	Mistral-7B (7.2B)	54.6	47.0 (↓ 7.6)	51.1	49.3	44.8	42.7
	Gemma-7B (8.5B)	53.5	47.8 (↓ 5.7)	51.7	50.1	47.6	41.8

Table 2: Main results of ATA's direct attacks on GSM8K (0-shot), BBH (3-shot), and MMLU (5-shot) for smaller models. Results expressed in accuracy (%). All models are chat models.

geted attack on the question. The overall algorithm is further illustrated in Algorithm 1.

3 Experiment

3.1 Experimental Setup

Dataset. For our experiments, we have selected three widely recognized reasoning datasets: GSM8K (Cobbe et al., 2021), BBH (Suzgun et al., 2023), and MMLU (Hendrycks et al., 2021), which cover evaluation of comprehensive reasoning capabilities, including logical reasoning, symbolic reasoning, mathematical reasoning, and commonsense reasoning. We include all test questions from the GSM8K dataset in our evaluation. For the BBH and MMLU datasets, due to computational constraints, we will select a subset of 50 questions from each topic.

Generation of adversarial test cases. We conduct ATA on both zero-shot and few-shot prompts, focusing specifically on editing the questions (and options, if applicable). Notably, we avoid attacking the standardized prompt, "Let's think step by step." to ensure the model retains its understanding of the need for CoT. For few-shot prompts, we retain the original examples without edits, simulating human behavior of directly copying examples.

Models. To evaluate the reasoning robustness of LLMs, we select LLMs ranging from smaller parameters to larger parameters to attack. We use Gemma-2B-It, Gemma-7B-It (Team et al., 2024), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), Llama2-7B-Chat (Touvron et al., 2023), Vicuna-13B-v1.5, Vicuna-33B-v1.3 (Chiang et al., 2023), Mixtral-8×7B-Instruct-v0.1 (Jiang et al., 2024), ChatGPT (gpt-3.5-turbo-0613) (OpenAI, 2022), GPT-4 (gpt-4-0613) (OpenAI, 2023). For the larger and closed-source models, such as Vicuna-33B-v1.3, Mixtral-8×7B-Instruct-v0.1, and ChatGPT, we employ questions generated by the smaller Mistral-7B-Instruct-v0.2 model to evaluate their performance. This approach demonstrates ATA's transferability across white-box models and between white-box and black-box models.

Implementation details. We present accuracy results for both the original and edited scores, represented on a logarithmic scale ranging from 1 to 8 edits applied to each question. The primary metric for assessing the effectiveness of an adversarial attack is the reduction in accuracy. All experiments are conducted on the A800-SMX-80GB GPU.

3.2 Main results

The main results of the attacks on the GSM8K, BBH, and MMLU datasets and comparison of the performance of the baselines models are summarized in Table 2 and Table 3.

Performance Degradation under ATA. As shown in Table 2 and Table 3, our method consistently reduces model performance across various datasets, demonstrating the significant vulnerability of LLMs to such errors. For instance, in Table 2, small models like Gemma-2B[2], Llama2-7B, Mistral-7B and Gemma-7B show striking average absolute reductions of 7.0%, 10.6%, 13.6% and 7.8% respectively for GSM8K. Similar declines are observed across four models on other datasets shown by 8.8%, 7.6%, 9.1%, and 6.5% respectively for BBH, and 6.6%, 5.6%, 7.6%, and 5.7%

[2] We will now use the abbreviated model name without the version information to avoid redundancy.

Dataset	Model (#Params)	Ori.	Avg-ATA	ATA-1	ATA-2	ATA-4	ATA-8
GSM8K	Vicuna-13B (13B)	33.4	28.4 (↓ 5.0)	32.4	30.8	26.2	24.3
	Vicuna-33B (33B)	38.2	29.2 (↓ 9.0)	35.3	32.6	26.4	22.5
	Mixtral-8×7B (47B)	68.5	60.9 (↓ 8.3)	66.7	62.8	57.9	53.4
BBH	Vicuna-13B (13B)	51.2	42.5 (↓ 8.7)	47.7	44.9	40.8	36.6
	Vicuna-33B (33B)	52.1	43.7 (↓ 8.4)	49.4	44.7	42.5	38.2
	Mixtral-8×7B (47B)	65.6	60.4 (↓ 5.2)	64.0	62.8	58.3	56.4
MMLU	Vicuna-13B (13B)	53.4	48.2 (↓ 5.2)	50.8	50.3	48.2	43.6
	Vicuna-33B (33B)	59.2	52.3 (↓ 6.9)	56.3	54.9	51.4	47.5
	Mixtral-8×7B (47B)	68.4	63.3 (↓ 5.1)	66.1	64.8	62.1	60.2

Table 3: Main results of transfer attacks on GSM8K (0-shot), BBH (3-shot), and MMLU (5-shot) for larger models. Adversarial data used to attack is from Mistral-7B. Results expressed in accuracy (%). All models are chat models.

Model	Method	GSM8K	BBH	MMLU	Avg.
Mistral-7B*	Original	43.7	50.0	56.6	50.1
	Random	39.2	48.4	54.8	47.5 (↓ 2.6)
	PromptBench	–	50.0	56.4	53.2 (↓ 0.1)
	ATA-4	27.1	39.1	48.3	38.2 (↓ 11.9)
Gemma-7B*	Original	39.9	42.4	53.5	45.3
	Random	40.3	41.2	53.4	45.0 (↓ 0.3)
	PromptBench	–	42.3	53.5	47.9 (↓ 0.1)
	ATA-4	29.8	33.5	47.6	37.0 (↓ 6.3)
Vicuna-33B$^+$	Original	38.2	52.1	59.2	49.8
	Random	37.4	52.2	57.9	49.2 (↓ 0.6)
	PromptBench	–	52.1	59.0	55.6 (↓ 0.1)
	ATA-4	26.4	42.5	51.4	40.1 (↓ 9.7)

Table 4: Performance compared to random selection and PromptBench, where * indicates direct applying ATA, while $^+$ indicates transfering from other models. Promptbench is not used to attack GSM8K dataset as there is no instruction used in GSM8K.

respectively for MMLU. These results consistently illustrate that even minor typographical errors can trigger significant performance degradation, reflecting a systemic weakness in LLMs' ability to handle imperfect input. The consistent decrease in accuracy across different datasets and models underscores the generalizability of our attack. By exploiting these vulnerabilities, our adversarial typographical errors disrupt the internal reasoning processes of LLMs, leading to erroneous outputs and highlighting a critical area for improvement for LLMs.

Transferability. To further explore the impact of adversarial typographical errors on LLMs, we evaluated the transferability of adversarial prompts crafted for Mistral-7B to larger models. The results reveal a similar vulnerability to smaller models, as larger models shown in Table 3: Vicuna-13B, Vicuna 33B, and Mixtral-8×7B show average absolute reductions of 5.0%, 9.0%, and 8.3% respectively for GSM8K, 8.7%, 8.4%, and 5.2% respectively for BBH, 5.2%, 6.9%, and 5.1% respectively for MMLU. This consistent decrease in performance across various larger models underscores the high transferability of our adversarial attacks, demonstrating that typographical errors not only disrupt smaller models but also significantly impair the reasoning processes of more complex systems. These findings emphasize that the vulnerabilities exploited by our attacks are fundamental, affecting a broad spectrum of model architectures and sizes, thereby highlighting the critical need for robust defense mechanisms in the development of future LLMs.

3.3 Attack Performance Analysis

Effectiveness. We compare ATA-4 with two baselines to evaluate its effectiveness. The first baseline, referred to as the random baseline, involves randomly choosing words and letters to be edited and replacing them by randomly sampling from a mistake dictionary. The second baseline employs the "DeepWordBug" strategy from Promptbench (Zhu et al., 2023), which targets the instruction portion of the prompts. As shown in Table 4, our results demonstrate that ATA-4 significantly outperforms both baselines in degrading model performance. For Mistral-7B, Gemma-7B, and Vicuna-33B, ATA-4 at 4 edits results in average absolute reductions in accuracy of 11.9%, 6.3%, and 9.7% respectively. In stark contrast, the random baseline yields much lower reductions of 2.6%, 0.3%, and 0.6%, while Promptbench's DeepWordBug strategy results in minimal reductions of 0.1%, 0.1%, and 0.1%. These findings underscore the superior effectiveness of ATA-4, which leverages targeted typographical errors to exploit model vulnerabilities more efficiently than random or instruction-focused attacks. This also demonstrates a clear and significant impact on the reasoning capabilities of LLMs compared to the

Model	**Task**	Ori.	ATA-1	ATA-2	ATA-4	ATA-8
ChatGPT$^+$	GSM8K	72 ± 0.8	68 ± 1.3	66 ± 2.5	62 ± 1.2	58 ± 1.7
	BBH	69 ± 0.4	68 ± 0.4	65 ± 0.7	61 ± 0.3	59 ± 0.6
	MMLU	67 ± 0.3	65 ± 0.2	63 ± 0.4	59 ± 0.6	56 ± 0.5
GPT-4$^+$	GSM8K	88 ± 0.5	87 ± 0.6	86 ± 0.5	84 ± 0.4	81 ± 0.7
	BBH	89 ± 0.6	89 ± 0.6	87 ± 0.7	86 ± 0.2	85 ± 0.6
	MMLU	86 ± 0.8	85 ± 0.4	84 ± 0.3	84 ± 0.9	82 ± 0.8

Table 5: Performance of ATA on closed-source models. ATA notably impacts ChatGPT but have a minimal impact on GPT-4, highlighting GPT-4's human-level comprehension and resistance to such errors. This affirms that ATA generates imperceptible typos in prompt.

baseline strategies.

Performance on ChatGPT and GPT-4. We conduct transfer experiments on ChatGPT and GPT-4. However, due to the high cost involved, we only sample 100 instances for each dataset, and we run for 3 times and report the results with their respective standard deviations in Table 5. ATA achieves an average performance drop of 8.5% on GSM8K, 5.8% on BBH, and 6.3% on MMLU. However, when targeting GPT-4, it fails to produce significant impact, resulting in an average performance drop of only 3.5% on GSM8K, 2.3% on BBH, and 2.3% on MMLU. The inability to attack GPT-4 demonstrates that when models possess a similar level of comprehension as humans, typos have negligible influence on the results. Moreover, this substantiates that ATA solely incorporates imperceptible typos within prompts.

4 Benchmark: Reasoning Robustness to Adversarial Typo Attacks (R^{2}ATA)

To enable a comprehensive evaluation of LLMs' Reasoning Robustness to ATA, including future new models, super-large models, and closed-source models, we propose the establishment of a benchmark named R^{2}ATA. This benchmark utilizes adversarial typographical questions derived from transfer experiments conducted in Section 3, specifically GSM8K, BBH, and MMLU. Concrete examples of R^{2}ATA for each dataset are shown in Tables 6 to 8 in Appendix A.1.

4.1 R^{2}ATA Statistics

Representative Example. Figure 3 compares the model's responses to an original and an adversarially edited GSM8K question. In the original question, the model follows a logical reasoning pathway to reach the correct answer. Meanwhile, the adversarially edited question introduces subtle typographical errors. These minor perturbations cause the model to misinterpret key terms, leading to erroneous intermediate steps and ultimately resulting in a wrong answer.

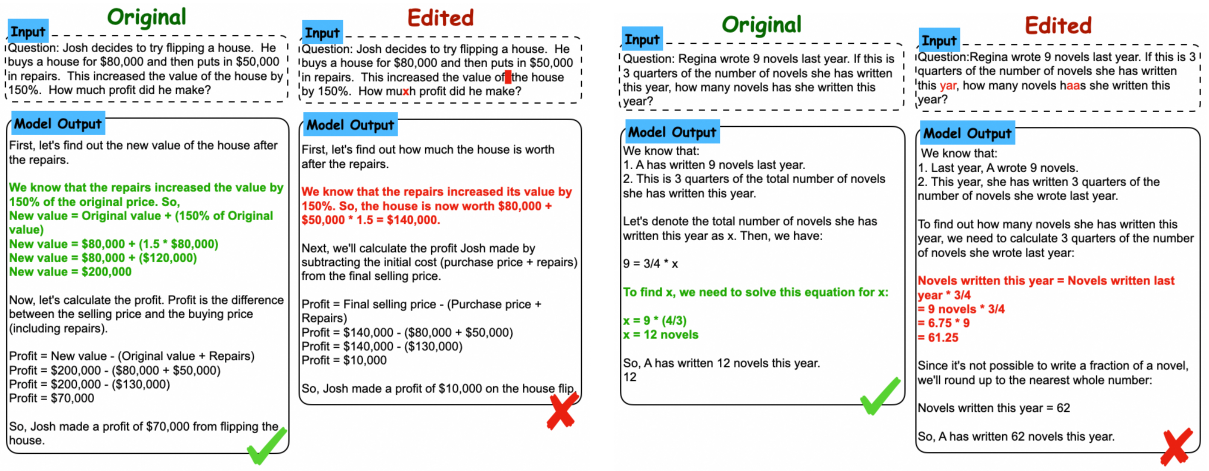
Distribution of Typographical Edits. One of the key analyses involves examining the distribution of the edit operations used in R^{2}ATA. Figure 4 illustrates the edit operation statistic present in R^{2}ATA. Notably, the predominance of the *whitespace* error operation adopted by ATA highlights its significance in exploiting model vulnerabilities. This suggests that LLMs are particularly susceptible to errors stemming from additional whitespace, possibly due to a lack of robustness in handling such perturbations. The frequency of whitespace errors implies that patterns involving multiple whitespaces between words are likely infrequent in the training data, resulting in heightened sensitivity and errors in reasoning outputs.

The variation in error operation distribution across the three datasets, as depicted in Figure 4, indicates that task complexity influences the prevalence of specific error operations. The GSM8K dataset focuses on mathematical reasoning, while MMLU and BBH cover a broader range of tasks, including logical and commonsense reasoning (Suzgun et al., 2023). By systematically evaluating LLMs' performance under these conditions, the benchmark aims to provide insights into improving model robustness across diverse reasoning tasks.

4.2 R^{2}ATA Analysis

The R^{2}ATA benchmark is analyzed at various levels to provide comprehensive insights into the types and patterns of typographical errors that impact model performance.

Type of Edited Words. Figure 5 illustrates the distribution of edited word types across all three datasets. The data reveals that nouns are the most frequently edited word type, accounting for 48.9% of the edits. Verbs follow at 16.7%, and adjectives at 14.9%. This distribution reflects the significant roles these word types play in conveying meaning. Nouns, as primary subjects and objects, are often targeted for edits due to their substantial semantic weight, which can profoundly alter sentence meaning and context. Verbs, crucial for actions and states, similarly impact sentence meaning when modified. Adjectives, providing descriptive nuances, can subtly change the tone or implication of text upon editing. In contrast, stop words such as



Question: Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?

First, let's find out the new value of the house after the repairs.

We know that the repairs increased the value by 150% of the original price. So,
New value = Original value + (150% of Original value)
New value = $80,000 + (1.5 * $80,000)
New value = $80,000 + ($120,000)
New value = $200,000

Now, let's calculate the profit. Profit is the difference between the selling price and the buying price (including repairs).

Profit = New value - (Original value + Repairs)
Profit = $200,000 - ($80,000 + $50,000)
Profit = $200,000 - ($130,000)
Profit = $70,000

So, Josh made a profit of $70,000 from flipping the house.

Question: Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs. This increased the value of█the house by 150%. How muxh profit did he make?

First, let's find out how much the house is worth after the repairs.

We know that the repairs increased its value by 150%. So, the house is now worth $80,000 + $50,000 * 1.5 = $140,000.

Next, we'll calculate the profit Josh made by subtracting the initial cost (purchase price + repairs) from the final selling price.

Profit = Final selling price - (Purchase price + Repairs)
Profit = $140,000 - ($80,000 + $50,000)
Profit = $140,000 - ($130,000)
Profit = $10,000

So, Josh made a profit of $10,000 on the house flip.

Question: Regina wrote 9 novels last year. If this is 3 quarters of the number of novels she has written this year, how many novels has she written this year?

We know that:
1. A has written 9 novels last year.
2. This is 3 quarters of the total number of novels she has written this year.

Let's denote the total number of novels she has written this year as x. Then, we have:

9 = 3/4 * x

To find x, we need to solve this equation for x:

x = 9 * (4/3)
x = 12 novels

So, A has written 12 novels this year.
12

Question:Regina wrote 9 novels last year. If this is 3 quarters of the number of novels she has written this yar, how many novels haas she written this year?

We know that:
1. Last year, A wrote 9 novels.
2. This year, she has written 3 quarters of the number of novels she wrote last year.

To find out how many novels she has written this year, we need to calculate 3 quarters of the number of novels she wrote last year:

Novels written this year = Novels written last year * 3/4
= 9 novels * 3/4
= 6.75 * 9
= 61.25

Since it's not possible to write a fraction of a novel, we'll round up to the nearest whole number:

Novels written this year = 62

So, A has written 62 novels this year.

(a) Whitespace and Replace Errors.

(b) Omission and Double.

Figure 3: Comparison of Mistral-7B responses to original (left) and adversarially edited (right) GSM8K questions. Minor typographical errors in the edited question can lead to misinterpretation and incorrect answers.

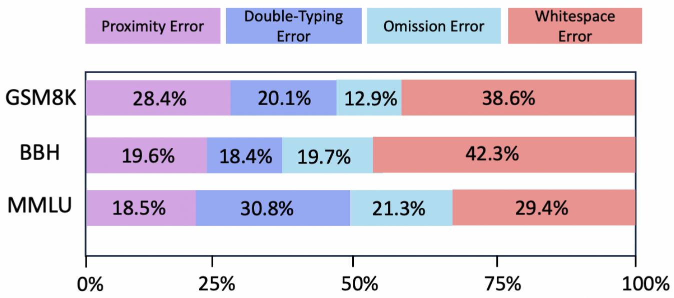


Figure 4: Distribution of error operations selected by ATA across the datasaets in R^{2}ATA benchmark. The predominance of whitespace errors highlights a key vulnerability in LLMs.

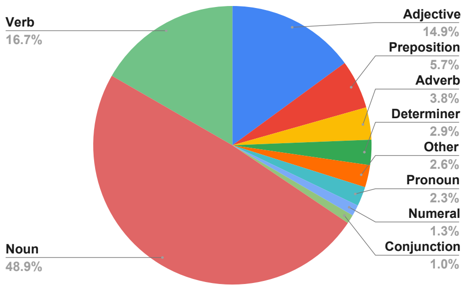


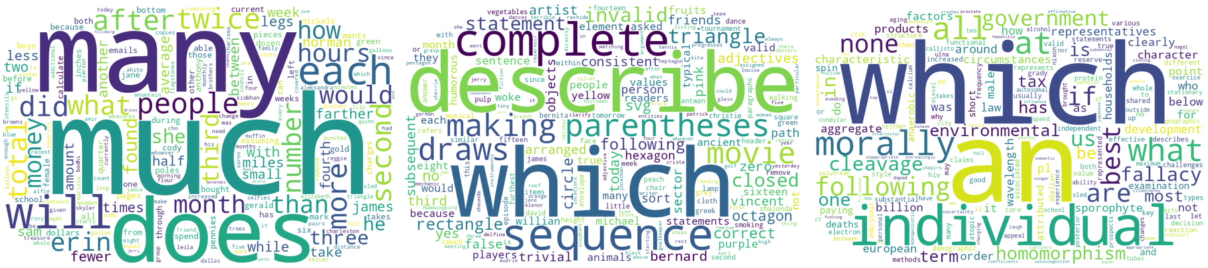
Figure 5: Distribution of edited word types in R^{2}ATA. Nouns, Verbs, and Adjectives constitute the majority of edited words.

conjunctions and prepositions primarily contribute to grammatical structure rather than semantic content, making them less frequently edited and thus less impactful on overall meaning. This goes to show that models need to be more robust to subject perturbations to ensure more robustness to these typographical errors.

Edited Words Statistics. Figure 6 shows the word cloud of edited words with size reflecting edit frequency. To ensure a fair comparison, we applied Inverse Document Frequency (IDF) normalization, calculated using: $\mathrm{IDF}(t) = \log\left(\frac{N}{df_t}\right)$, where t is the term, N is the total number of prompts, and df_t is the number of prompts containing the term t.

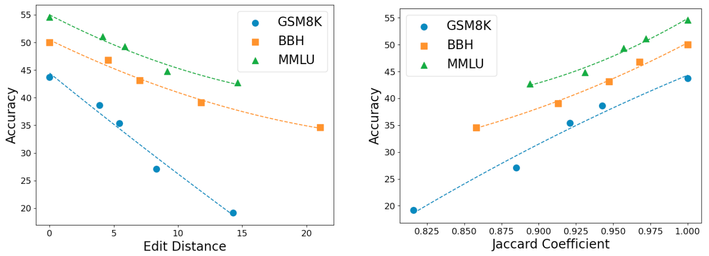
We adjust each word's frequency by multiplying it with its IDF weight to highlight words disproportionately edited relative to their overall frequency. In the GSM8K dataset, frequent edits of words like "many," "people," "much," "two," "each," and "total" suggest their semantic importance in mathematical problems due to their inherent complexity and the model's sensitivity to linguistic patterns and numerical expressions. Figures 6(b) and 6(c) show word clouds from BBH and MMLU datasets, highlighting words like "describe," "which," "complete" for BBH, and "individual," "an," "which," "all," and "morally" for MMLU, which cover diverse topics compared to GSM8K's focus on math. The minimal presence of stop words among frequently edited words indicates that edits target content-bearing words, suggesting that ATA edits aim to disrupt the text's logical flow, coherence, or semantics, thus strategically influencing the model's reasoning abilities.

Impact on the Token Level. Figure 7a illustrates the how accuracy varies with edit distance for adversarially edited prompts across three datasets: GSM8K, BBH, and MMLU. Meanwhile, Figure 7b shows how accuracy varies with the Jaccard coefficient, with each data point representing 0, 1,



(a) GSM8K (b) BBH (c) MMLU

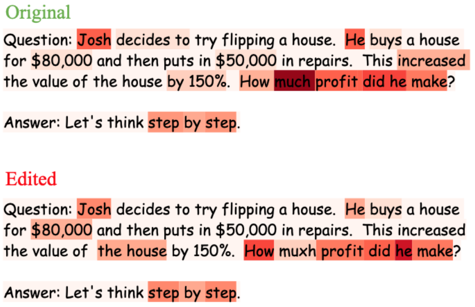
Figure 6: Statistic of words edited in R^{2}ATA.



(a) Edit Distance. From left to right, each data point represents 0, 1, 2, 4, 8 edits respectively.

(b) Jaccard Coefficient. From left to right, each data point represents 8, 4, 2, 1, 0 edits respectively.

Figure 7: Effects of adversarial edits at the token level.


Original

Question: Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?

Answer: Let's think step by step.

Edited

Question: Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs. This increased the value of the house by 150%. How muxh profit did he make?

Answer: Let's think step by step.

Figure 8: Visualizing attention changes before and after adversarial attacks.

2, 4, and 8 edits. It is evident that even a small number of edits leads to a substantial increase in edit distance, resulting in a significant decline in accuracy. However, despite this increase in edit distance, the Jaccard coefficient remains relatively stable, consistently exceeding 0.8 across all edits. This high degree of similarity between the edited and original prompts suggests that the edits are likely imperceptible to humans, underscoring the challenge of detecting adversarial modifications.

Impact on Attention Figure 8 illustrates the changes in attention distribution before and after an adversarial attack on a question. In the original question, attention was focused on critical words such as "much," "increased," and "by 150%". However, after the question was edited, there was a noticeable shift in attention. For instance, the attention on "much" decreased significantly due to it being altered to "muxh". Similarly, attention on "increased" and "by 150%" was entirely lost. Instead, the attention was redirected to irrelevant words like "the house". This misallocation of attention led to errors in the reasoning steps, as the model focused on less important parts of the text, thereby compromising its ability to understand and answer the question correctly. The detailed implementation code for attention calculation using PyTorch is shown in Appendix A.2.

5 Related Work

Textual Adversarial Attacks have garnered significant attention due to their ability to exploit vulnerabilities in LLMs. These attacks, which manipulate input text to mislead models into incorrect predictions or misleading responses, have been studied extensively at various levels of input granularity: character-level (Gao et al., 2018; Li et al., 2019; Pruthi et al., 2019), word-level (Garg and Ramakrishnan, 2020; Jin et al., 2020; Zhou et al., 2024), sentence-level (Shi et al., 2023; Xu et al., 2024; Turpin et al., 2024; Lanham et al., 2023) and semantic-level Zhu et al. (2023); Parcalabescu and Frank (2023), as noted by Zhu et al. (2023). However, these approaches often generate adversarial examples that are easily detectable by human users, limiting their real world applicability. Our approach instead introduces imperceptible modifications to prompts similar to Brown et al. (2018); Richards et al. (2021), offering a more realistic assessment of adversarial risks.

Furthermore, while some defenses address related threats, such as malware detection adversaries (Fleshman et al., 2018; Íncer Romeo et al., 2018), they operate in more constrained spaces and do not directly apply to the nuanced edits (Lowd and Meek, 2005) that we explore.

6 Conclusion

This study examined the robustness of LLMs to typographical errors using the ATA algorithm and the R^{2}ATA benchmark. By focusing on imperceptible, real-world attacks in NLP, our work fills a key gap in adversarial research, moving beyond the artificial constrains of prior approaches and offering insights into more practical vulnerabilities in LLMs. Our findings show that even minor typographical changes significantly reduce model accuracy. Specifically, we observe that adversarial prompts from Mistral-7B similarly affect larger models like Vicuna-13B, Vicuna-33B, and Mixtral-8×7B, indicating that both smaller and larger models are vulnerable. This highlights the need for improved robustness in LLMs against typographical errors. The R^{2}ATA benchmark is a valuable tool for developing more resilient models capable of reliable performance despite minor errors, emphasizing the critical need for robust defense mechanisms in future LLMs.

Limitation

Our algorithm primarily focuses on typographical errors common in languages that use alphabets and whitespaces, such as English. This excludes languages with different writing systems, such as Chinese, where typographical errors may involve character substitutions or stroke omissions. The typographical errors considered may not cover all possible real-world scenarios. For instance, whitespace errors only apply to languages that use spaces, while letter addition and deletion errors are relevant only to alphabetic languages. Therefore, future research should extend the scope to encompass a broader range of linguistic diversity to ensure the applicability of findings across various languages and writing systems. Exploring language-specific modifications will provide a more comprehensive understanding of LLM robustness across diverse linguistic contexts. Developing and testing adversarial attacks tailored to these languages will help in creating more universally resilient language models. Additionally, our evaluation primarily relies on open-source and commercially available LLMs due to accessibility constraints. While the R^{2}ATA benchmark effectively demonstrates vulnerabilities in these models, the performance of many closed-source LLMs remains unexplored.

Acknowledgements

This research is partially supported by the National Research Foundation Singapore under the AI Singapore Programme (AISG Award No: AISG2-TC-2023-010-SGIL) and the Singapore Ministry of Education Academic Research Fund Tier 1 (Award No: T1 251RES2207). Additionally, the computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore.

References

Tom B. Brown, Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Christiano, and Ian Goodfellow. 2018. Unrestricted adversarial examples.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

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. *arXiv preprint arXiv:2110.14168*.

William Fleshman, Edward Raff, Jared Sylvester, Steven Forsyth, and Mark McLean. 2018. Non-negative networks against adversarial attacks. *arXiv preprint arXiv:1806.06108*.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181.

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

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *Proceedings 2019 Network and Distributed System Security Symposium*, NDSS 2019. Internet Society.

Daniel Lowd and Christopher Meek. 2005. Good word attacks on statistical spam filters. In *CEAS*, volume 2005.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2023. Gpt-4 technical report.

Letitia Parcalabescu and Anette Frank. 2023. On measuring faithfulness of natural language explanations. *arXiv preprint arXiv:2311.07466*.

Fábio Perez and Ian Ribeiro. 2022. Ignore previous prompt: Attack techniques for language models. In *NeurIPS ML Safety Workshop*.

Danish Pruthi, Bhuwan Dhingra, and Zachary C Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5582–5591.

Luke E. Richards, André Nguyen, Ryan Capps, Steven Forsyth, Cynthia Matuszek, and Edward Raff. 2021. Adversarial transfer attacks with unknown data and class overlap. In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, AISec '21, page 13–24, New York, NY, USA. Association for Computing Machinery.

Íñigo Íncer Romeo, Michael Theodorides, Sadia Afroz, and David Wagner. 2018. Adversarially robust malware detection using monotonic classification. In *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics*, IWSPA '18, page 54–63, New York, NY, USA. Association for Computing Machinery.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2024. Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. 2024. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. *Advances in Neural Information Processing Systems*, 36.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhen Xiang, Fengqing Jiang, Zidi Xiong, Bhaskar Ramasubramanian, Radha Poovendran, and Bo Li. 2024. Badchain: Backdoor chain-of-thought prompting for large language models. In *NeurIPS 2023 Workshop on Backdoors in Deep Learning - The Good, the Bad, and the Ugly*.

Xilie Xu, Keyi Kong, Ning Liu, Lizhen Cui, Di Wang, Jingfeng Zhang, and Mohan Kankanhalli. 2024. An

LLM can fool itself: A prompt-based adversarial attack. In *The Twelfth International Conference on Learning Representations*.

Yunxiang Zhang, Liangming Pan, Samson Tan, and Min-Yen Kan. 2022. Interpreting the robustness of neural NLP models to textual perturbations. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3993–4007, Dublin, Ireland. Association for Computational Linguistics.

Zihao Zhou, Qiufeng Wang, Mingyu Jin, Jie Yao, Jianan Ye, Wei Liu, Wei Wang, Xiaowei Huang, and Kaizhu Huang. 2024. Mathattack: Attacking large language models towards math solving ability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19750–19758.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. 2023. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*.

A Appendix

A.1 Examples of Edits by ATA

GSM8K Table 6 shows an example of typographical errors generated on a GSM8K question.

Original	**Question: Archie buys beef jerky that comes 30 sticks to a bag and costs $18.00 per bag. If Archie buys 1 bag while they're $3.00 off a bag, how much will each stick of jerky cost in cents?\nAnswer: Let's think step by step.**
ATA-1	Question: Archie buys beef jerky that comes 30 sticks to a bag and costs $18.00 per bag. If Archie buys 1 bag while they're $3.00 off a bag, how much will each stick of jerky cost in cents?\nAnswer: Let's think step by step.
ATA-2	Question: Archie buys beec jerky that comes 30 sticks to a bag and costs $18.00 per bag. If Archie buys 1 bag while they're $3.00 off a bag, how much will each stick of jerky cost in cents?\nAnswer: Let's think step by step.
ATA-4	Question: Archie buys beec jerky that comes 30 sticks to a bag and costs $18.00 per bag. If Archie buys 1 bag while they're $3.00 off a bag, how much will each stick of jer kg cost in cents?\nAnswer: Let's think step by step.
ATA-8	Question: Archie buys beec jerky that comes 30 sticks to a bag and costs $18.00 per bag. If Archie buys 1 bag while they're $3.00 off a bag, how much will eacn stick of jer kg cost in cents?\nAnswer: Let's think step by step.

Table 6: An example of typographical errors generated on a GSM8K question.

BBH Table 7 shows an example of typographical errors generated on a BBH question.

MMLU Table 8 shows an example of typographical errors generated on a MMLU question.

Original	**Q: Is the following sentence plausible? "Petr Cech was safe at first."\nA: Let's think step by step.**
ATA-1	Q: Is the following sentence plausible? "Petr Cech was szfe at first."\nA: Let's think step by step.
ATA-2	Q: Is the following sentence plausible? "Petr Cech was szfe at first."\nA: Let's think step by step.
ATA-4	Q: Is the folllwing sentence plausible? "Petr Cech was szfe at first."\nA: Let's think step by step.
ATA-8	Q: Is the follwing sntence plausible? "Petr Cech was szfe at firsst."\nA: Let's think step by step.

Table 7: An example of typographical errors generated on a BBH question.

Original	**Q: Which of these should an objective NOT be?\n(A) Broad (B) Achievable (C) Measurable (D) Time-bound\nA: Let's think step by step.**
ATA-1	Q: Which of these should an objective NOT be?\n(A) Broad (B) Achievable (C) Measurable (D) Tie-bound\nA: Let's think step by step.
ATA-2	Q: Which of these should an objective NOT be?\n(A) Brod (B) Achievable (C) Measurable (D) Tie-bound\nA: Let's think step by step.
ATA-4	Q: Which of these should an object ve NOT be?\n(A) Brod (B) Achievable (C) Measurable (D) Tie-bound\nA: Let's think step by step.
ATA-8	Q: Which of thee shoulld an object ve NOT be?\n(A) Brod (B) Achievable (C) Me aaurable (D) Tie-bound\nA: Let's think step by step.

Table 8: An example of typographical errors generated on a MMLU question.

A.2 Calculation of Attention Weights

We obtained the attention weights using the Huggingface library. We obtain from specifically the last attention layer. Because there are 16 attention heads, we chose to perform mean pooling on the attention weight matrix and obtained the attention of all the words with respect to the last token in the user input.

```
from transformers import AutoModelForCausalLM
from transformers import AutoTokenizer


model = AutoModelForCausalLM.from_pretrained(
        model_name,output_attentions=True)
tokenizer = AutoTokenizer.from_pretrained(model_name)

messages = [
    {"role": "user", "content": "Question: Josh..."}
]

inputs = tokenizer.encode(messages,
        return_tensors='pt')
input_ids = inputs['input_ids']

attention = model(input_ids,attn_output_weights=True)
attention_last = attention_all[-1].mean()
```