Robustness Over Time: Understanding Adversarial Examples' Effectiveness on Longitudinal Versions of Large Language Models

Yugeng Liu*, Tianshuo Cong*, Zhengyu Zhao, Michael Backes, *Fellow, IEEE,* Yun Shen, Yang Zhang, *Member, IEEE,*

***Abstract*—Large Language Models (LLMs) undergo continuous updates to improve user experience. However, prior research on the security and safety implications of LLMs has primarily focused on their specific versions, overlooking the impact of successive LLM updates. This prompts the need for a holistic understanding of the risks in these different versions of LLMs. To fill this gap, in this paper, we conduct a longitudinal study to examine the adversarial robustness – specifically misclassification, jailbreak, and hallucination – of three prominent LLM families: GPT, Llama, and Qwen. Our study reveals that LLM updates do not consistently improve adversarial robustness as expected. For instance, a later version of GPT-3.5 degrades regarding misclassification and hallucination despite its improved resilience against jailbreaks. GPT-4 and GPT-4o demonstrate (incrementally) higher robustness overall. Larger Llama and Qwen models do not uniformly exhibit improved robustness across all three aspects studied. In addition, larger model sizes do not necessarily yield improved robustness. Minor updates lacking substantial robustness improvements can exacerbate existing issues rather than resolve them. We hope our study can offer valuable insights into navigating model updates and informed decisions in model development and usage.**

***Index Terms*—Robustness, Large Language Model, Adversarial Examples, Jailbreak, Hallucination.**

I. Introduction

LARGE Language Models (LLMs), such as GPT models by OpenAI [1], [2], Llama by Meta [3]–[5], and Qwen [6]–[9] by Alibaba, have demonstrated remarkable capabilities in varied Natural Language Processing (NLP) tasks, including language translation [10], text classification [11], and creative writing [12]. Despite their impressive performance, the outputs generated by LLMs can perpetuate harmful stereotypes [13]–[15], disseminate false information [16]–[20], or produce inappropriate and offensive responses [21]. In response, updates have been made to improve LLMs by incorporating feedback and insights from users and developers.

Although such improvements partially mitigate known attacks or failures observed in earlier versions [21], [22], unintended consequences and even new vulnerabilities or biases could still be introduced. Furthermore, in the era of LLMs, as the capabilities of models have further advanced, the robustness of generative models also involves building chatbot applications that execute complex user instructions by integrating tools, such as jailbreak and hallucination attacks. Unfortunately, current research on the robustness evaluation of LLMs has focused on a *single version* of the LLM and lacks a *holistic analysis* of these new adversarial examples, leaving the impact of model updates unexplored.

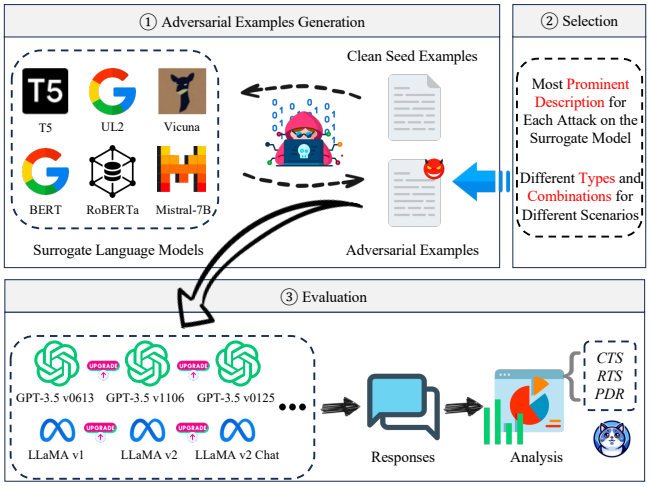


Fig. 1: Overview of our evaluation framework on adversarial robustness of LLMs over time using adversarial examples generated from various surrogate models.

Methodology. To fill this gap, in this paper, we undertake the first comprehensive robustness evaluation of longitudinal LLMs. We focus on assessing the robustness of popular LLMs over time, including closed-source OpenAI models (GPT-3.5/4/4o) as well as the open-source models (Llama 1/2/3 and Qwen 1.5/2/2.5/3). We utilize adversarial examples within the in-context learning (ICL) framework to examine this robustness [23]. Our evaluation workflow, illustrated in Figure 1, first generates adversarial examples using surrogate language models (e.g., T5 [24] or Mistral-7B [25]). These adversarial inputs are then tested against different versions of the target LLM. This allows us to examine how these adversarial inputs impact various versions of LLMs over time.

Remark. Following software engineer practice, we distinguish between model "upgrade" and "update" for fine-granularity

The first two authors made equal contributions.
Yugeng Liu, Michael Backes, and Yang Zhang are with the CISPA Helmholtz Center for Information Security, 66123 Saarbrucken, Germany. E-mail: yugeng.liu, director, zhang@cispa.de
Tianshuo Cong is with the Tsinghua University No. 30 Shuangqing Road, Haidian District, Beijing 100084, China E-mail: congtianshuo@gmail.com
Zhengyu Zhao is with Xi'an Jiaotong University, No. 28, Xianning West Road, Xi'an, Shaanxi 710049, China. E-mail: zhengyu.zhao@xjtu.edu.cn
Yun Shen is with Flexera, The Capitol Building, Oldbury, Bracknell, Berkshire, United Kingdom. E-mail: yun.shen@flexera.com

analysis. An *LLM upgrade* denotes a significant version change or major improvement, while an *LLM update* typically incorporates enhancements to the existing version. For example, the release of the `gpt-3.5-turbo-0125` (GPT-3.5 v0125) represents an upgrade to the `gpt-3.5-turbo-1106` (GPT-3.5 v1106). We consider `gpt-3.5-turbo` as a continuously updated model.

Findings. Firstly, we demonstrate no significant robustness improvement for LLMs during the *upgrade* process. For instance, `gpt-3.5-turbo-1106` (GPT-3.5 v1106), despite its better performance in jailbreak tests, shows the worst performance in both misclassification and hallucination evaluations compared to `gpt-3.5-turbo-0613` (GPT-3.5 v0613) and `gpt-3.5-turbo-0125` (GPT-3.5 v0125). For the Llama and Qwen families, upgrading models does not improve robustness in many cases. The overall performance of the Llama-3 families is much worse in all the tasks. Moreover, our findings reveal that increasing LLM size does not guarantee enhanced robustness, especially for Llama families. Qwen models are more vulnerable to adversarial questions than other adversarial content. **Secondly**, through the lens of misclassification evaluation, we conduct weekly tests on the GPT-3.5 and GPT-4 models to evaluate the minor update process. Our experiments show that model providers can update LLMs to incorporate new features over time. In general, these updates could nonetheless affect the overall performance of the LLMs, potentially exacerbating existing issues rather than resolving them.

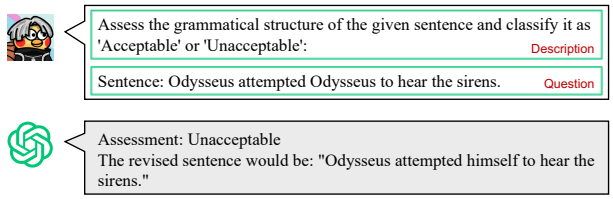
Impact. Our findings suggest that LLM upgrades or updates do not inherently enhance robustness. LLM users should prioritize robustness evaluation before deployment to avoid unexpected consequences. LLM developers must implement robustness strategies and thorough evaluations to fortify LLM resilience against unforeseen challenges.

II. Threat Modeling

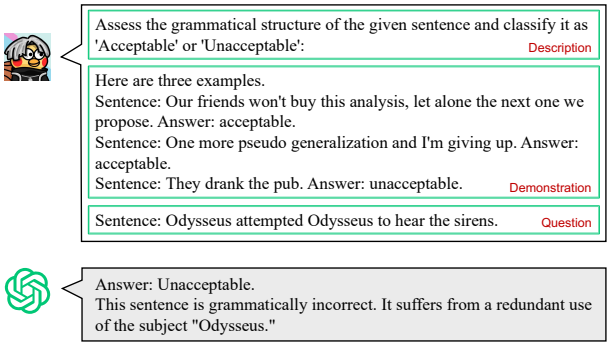
LLMs can perform completely different tasks, such as answering questions, summarizing documents, translating languages, and completing sentences. Within these tasks, according to different requirements and purposes, we categorize them into two types:

- **Classification Task.** LLMs are usually used to classify the input sentences, such as sentiment analysis (positive or negative) and linguistic acceptability (acceptable or not). These tasks usually require users to provide clear descriptions and corresponding candidate labels. Then, LLMs generate the prediction label to assist users.
- **Generation Task.** In this task, LLMs typically need to produce "open-ended" text based on the inputs from the user, with the goal of creating coherent, meaningful, and contextually appropriate answers. These answers are obtained from pre-trained knowledge of LLMs or some external searches.

Since in-context learning is the most widely used inference framework on both classification tasks and generation tasks, we inject adversarial examples into the components of in-context learning to evaluate the robustness of continuously updated LLMs. In this paper, we summarize all the queries into two types of in-context learning methods, i.e., zero-shot learning and few-shot learning.



(a) Q_{zero}



(b) Q_{few}

Fig. 2: Examples of (a) zero-shot learning and (b) few-shot learning on GPT-3.5 in the benchmark datasets. For zero-shot learning, the query includes only the description and the question but without any demonstrations, while few-shot learning means that the query also includes a few demonstrations.

A. In-context Learning (ICL)

Overview. The core idea of In-context Learning (ICL) is to learn from analogies implied in contextual information. ICL requires a few examples to form a demonstration context and feed them into LLM. It does not modify the parameters of an LLM and relies on the model to learn the pattern hidden in the demonstration (and accordingly generate the correct output). As such, ICL effectively reduces the computation costs for adapting LLMs to new tasks, as fine-tuning is not required. In general, there are two categories of in-context learning: zero-shot learning and few-shot learning. We outline their details as follows, and their examples can be found in Figure 2.

Zero-shot Learning. Zero-shot learning [26] is a capability enabled by LLMs, allowing them to generalize to tasks or domains they have never been explicitly trained on, no matter the classification or generation task. As a special case of ICL, the query of zero-shot learning to an LLM (termed Q_{zero} in this paper) only contains two elements: description and question (see Figure 2a), which can be formulated as follows.

$$Q_{zero} = \textit{Description} + \textit{Question}. \quad (1)$$

The *Description* guides the LLM, offering task details and response formats, while the *Question* defines the task. Zero-shot learning relies on the ability of an LLM to infer input and output from a query without demonstrations [27]. Thus, LLMs can perform well on unfamiliar tasks, making zero-shot learning a convenient, commonly used approach.

TABLE I: GPT family model list for the evaluation.

Model Family	Model Name	Short Name
GPT-3.5	gpt-3.5-turbo-0613	GPT-3.5 v0613
	gpt-3.5-turbo-1106	GPT-3.5 v1106
	gpt-3.5-turbo-0125	GPT-3.5 v0125
GPT-4	gpt-4-0613	GPT-4 v0613
	gpt-4-1106-preview	GPT-4 v1106
	gpt-4-0125-preview	GPT-4 v0125
	gpt-4-turbo-2024-04-09	GPT-4 v0409
GPT-4o	gpt-4o-05-13	GPT-4o v0513
	gpt-4o-2024-08-06	GPT-4o v0806
	gpt-4o-2024-11-20	GPT-3.5 v1120)

TABLE II: Llama family model list for the evaluation.

Model Family	Model Name	Short Name
Llama-7B	llama-7b	Llama-7B v1
	llama-2-7b	Llama-7B v2
	llama-2-7b-chat	Llama-7B v2C
	llama-3-8b	Llama-8B v3
	llama-3-8b-instruct	Llama-8B v3I
Llama-13B	llama-13B	Llama-13B v1
	llama-2-13b	Llama-13B v2
	llama-2-13b-chat	Llama-13B v2C
Llama-70B	llama-65B	Llama-65B v1
	llama-2-70b	Llama-70B v2
	llama-2-70b-chat	Llama-70B v2C
	llama-3-70b	Llama-70B v3
	llama-3-70b-instruct	Llama-70B v3I

TABLE III: Qwen family model list for the evaluation.

Model Family	Model Name	Short Name
Qwen-7B	qwen1.5-7b-chat	Qwen-7B v1.5C
	qwen2-7b-instruct	Qwen-7B v2I
	qwen2.5-7b-instruct	Qwen-7B v2.5I)
	qwen3-8b	Qwen-8B v3
Qwen-32B	qwen1.5-32b-chat	Qwen-32B v1.5C
	qwen2.5-32b-instruct	Qwen-32B v2.5I
	qwen3-32b	Qwen-32B v3
Qwen-72B	qwen1.5-72b-chat	Qwen-72B v1.5C
	qwen2-72b-instruct	Qwen-72B v2I
	qwen2.5-72b-instruct	Qwen-72B v2.5I

Few-shot Learning. Few-shot learning [23] includes a few examples to form a learning context that enables LLMs to better understand the task. In other words, compared to zero-shot learning, few-shot learning enables LLMs to quickly adapt to new tasks by learning from an extra element, i.e., *Demonstration*. Thus, the query of few-shot learning, Q_{few}, can be formulated as:

$$Q_{few} = \textit{Description} + \textit{Demonstration} + \textit{Question}. \quad (2)$$

Demonstration typically includes question-answer pairs, either labels in classification or semantically similar examples in generation tasks. An example of Q_{few} for classification is shown in Figure 2b. Few-shot learning helps LLMs form more accurate mappings between questions and answers. In this paper, following [28], we focus on 3-shot learning (three question-answer pairs in the *Demonstration*).

B. Over Time

In this paper, "over time" pertains to the target LLMs that undergo continuous *upgrades* and *updates* under the direction of their developers, which will be explained as follows.

Upgrade Over Time. Recent studies have shown that adversarial examples possess significant transferability across different LLMs [28]. However, many LLMs, like ChatGPT and Llama, undergo continuous upgrades. This raises the question of whether these successive versions remain vulnerable to prior adversarial strategies, necessitating a systematic evaluation of the latest LLM iterations. We conduct a comprehensive robustness assessment of GPT, Llama, and Qwen models that are subject to ongoing updates. We list the specific models and the short names from three families in Table I, Table II, and Table III, respectively.

More specifically, we treat `llama-3-8b` as an upgrade from `llama-7b`, and `llama-2-7b` as well as `llama-2-70b` and `llama-3-70b` as successors to `llama-65b`, as well as Qwen models. Following the structures of Q_{zero} and Q_{few} (see Section II-A), we introduce different adversarial examples into these longitudinal LLM versions to assess robustness.

Update Over Time. In addition to major upgrades, closed-source commercial LLMs often undergo minor updates within a single version. OpenAI informs users of model updates via email, but some updates occur without notification. Thus, we assert that regular updates for the current versions are necessary. To examine robustness over time, we utilize the latest iterations of GPT-3.5 (now `gpt-3.5-turbo-0125`)[1] and GPT-4 (now `gpt-4-0613`) as our target models. As in previous upgrade experiments, we introduce various adversarial examples into GPT-3.5 and GPT-4 to assess robustness, conducting weekly evaluations from the fourth week of 2024.

We also report our attack taxonomy in Section III and evaluation setup, such as our datasets (see Section IV-A) and the details of our evaluation metrics (see Section IV-B).

III. Attack Taxonomy

In this section, we primarily discuss the taxonomy of adversarial attacks on LLMs. The goal of adversarial attacks is to manipulate the behavior of LLMs to elicit misleading or undesirable responses through adversarial examples in *clean* queries [28]–[30]. We consider three adversarial attacks against LLMs. The first is *misclassification* (see Section III-A), which tries to perturb normal input to induce the model to deviate from correct predictions. In addition, a more representative adversarial attack is *jailbreak* (see Section III-B). Jailbreak attacks aim to modify the input to bypass the safeguard within LLMs, thereby forcing LLMs to generate certain disallowed, unsafe, or harmful statements. Furthermore, another adversarial attack, named *hallucination* (see Section III-C), is to produce coherent and grammatically correct but factually incorrect or nonsensical outputs.

A. Misclassification

Previous studies have proposed various effective methods to generate adversarial examples against language models. However, different works consider attacking different components of the ICL framework. For instance, Zhu et al. [28] proposed PromptBench, a dataset consisting of adversarial descriptions. Their experimental results demonstrate that descriptions are notably vulnerable to adversarial attacks primarily because

[1]Prior to February 16th, 2024, GPT-3.5 pointed to `gpt-3.5-turbo-0613`.

TABLE IV: Instances of Adversarial Description and Adversarial Question on SST-2 task.

Element	Type	Instances
Description	clean	Evaluate the sentiment of the given text and classify it as 'positive' or 'negative':
	Adversarial	Evaluate the sentiment of the given text and classify it as 'positive' or 'negative' 5yWbBXztUY:
Question	clean	Some actors have so much charisma that you 'd be happy to listen to them reading the phone book.
	Adversarial	Some actors have so much charisma that you 'd be jovial to listen to them reading the phone book.

they serve as a critical context that shapes the responses of LLMs and guides their cognitive outputs. However, they only focus on a single version of LLMs. Wang et al. [31] proposed AdvGLUE, a dataset comprising adversarial questions, but the target models are traditional small-scale language models. Above all, to fully evaluate the robustness of the LLMs, we consider different types of adversarial queries, i.e., each element of a query can be clean or adversarial. The clean and adversarial examples of each element are shown in Table IV.

Zero-shot Learning. We first outline different categories of queries in zero-shot learning under misclassification (mc) task, i.e., Q^{mc}_{zero}. Recall that zero-shot learning does not have any demonstration; thus, Q_{zero} encompasses two elements: the description and the question. We can replace any of them with Adversarial (A) or Clean (C) examples. For instance, we use Q^{AC}_{zero} to denote an adversarial query consisting of an adversarial description and a clean question. In turn, we can generate the clean or adversarial queries Q^{mc}_{zero} as follows:

$$Q^{\mathsf{mc}}_{zero} := \{Q^{\mathsf{CC}}_{zero}, Q^{\mathsf{AC}}_{zero}, Q^{\mathsf{CA}}_{zero}, Q^{\mathsf{AA}}_{zero}\}.$$

Few-shot Learning. Similar to the procedure employed for Q^{mc}_{zero}, we extend our approach to encompass the creation of adversarial queries in few-shot learning, i.e., Q^{mc}_{few}. For instance, Q^{AAC}_{few} consists of an adversarial description, an adversarial demonstration, and a clean question. Given Q^{mc}_{few} with several demonstrations, we can generate the clean or adversarial queries in eight scenarios as follows:

$$Q^{\mathsf{mc}}_{few} := \begin{Bmatrix} Q^{\mathsf{CCC}}_{few}, Q^{\mathsf{ACC}}_{few}, Q^{\mathsf{CAC}}_{few}, Q^{\mathsf{CCA}}_{few}, \\ Q^{\mathsf{AAC}}_{few}, Q^{\mathsf{ACA}}_{few}, Q^{\mathsf{CAA}}_{few}, Q^{\mathsf{AAA}}_{few} \end{Bmatrix}.$$

B. Jailbreak

Jailbreak attacks refer to a scenario where a user intentionally tries to trick or manipulate the LLMs to bypass their built-in safety, ethical, or operational guidelines, thereby inducing the LLMs to produce toxic responses. Previous works [32]–[36] have proposed a new class of adversarial attacks that can jailbreak safety-aligned language models. Among them, to test the robustness of both closed-source and open-source LLMs against jailbreak attacks, we launch black-box *optimization-based* jailbreak attacks, including GPTfuzz [33], PAIR [35], and TAP [36]. More concretely, these methods are systematically automated prompt-level jailbreaks optimized by outputs or coordinates of LLMs. Note that all of the above methods are zero-shot learning. We attempt to use few-shot learning for jailbreaking as well. However, the results indicate that once an answer in the demonstration involves a jailbreaking response, the LLM is more likely to reject the query, regardless of the optimization method used. Therefore, in this paper, we do not discuss few-shot learning. In addition, beyond the scope of jailbreak queries, the method for adversarial-based techniques needs a detailed description for modification. More specifically, a surrogate or teacher model leverages the outputs of the target LLM to refine or alter the description. Consequently, within our framework, we assume that the question itself invariably originates from a clean source, but the descriptions are classified as either clean or adversarial, as delineated in Equation 1. Given the adversarial query Q^{jb}_{zero}, we have the following adversarial queries in two scenarios:

$$Q^{\mathsf{jb}}_{zero} := \{Q^{\mathsf{CC}}_{zero}, Q^{\mathsf{AC}}_{zero}\}.$$

In this paper, the adversary is assumed to have black-box access to the LLMs for the following three adversarial-based optimized jailbreak attacks.

- **GPTfuzz [33]** To automate the generation of jailbreak templates for LLMs, GPTfuzz starts with human-written templates. It uses a series of random mutations to generate new inputs and evaluate them with the assistance of LLMs.
- **PAIR [35]** PAIR is a systematically automated prompt-level jailbreak. PAIR adopts an *attacker* LLM to discover and improve the jailbreak prompts and uses a *judge* LLM to evaluate the responses from the *target* LLM.
- **TAP [36]** More advanced than PAIR, TAP utilizes three LLMs: an *attacker*, an *evaluator*, and a *target*. The task is to generate the jailbreaking prompts using tree-of-thoughts reasoning. The *evaluator* first assesses the generated prompts and evaluates whether the jailbreaking attempt would be successful or not, and then evaluates the generated prompts from the *target*.

C. Hallucination

Hallucination is commonly defined as a phenomenon in which a model generates incorrect, nonsensical, or imaginary content in unconstrained generation settings. In this work, however, we do not study hallucination as a free-form generation behavior. Instead, we focus on hallucinated answers as adversarial examples and investigate the robustness of LLMs when explicitly confronted with such inputs. Specifically, we treat hallucination answers as adversarially constructed alternatives that introduce logical inconsistencies and misleading evidence, rather than linguistic or syntactic errors. This formulation allows us to define a controlled robustness-oriented threat model, where the objective is to assess whether an LLM can resist being misled by hallucinated content. Since LLMs are often deployed as open-domain systems and act as repositories of world knowledge, robustness against hallucinated adversarial inputs is a critical reliability requirement.

To evaluate this robustness, we construct hallucinated answers across three domains: Question Answering (QA), Dialogue, and Summarization. Each query is paired with one correct answer and one hallucinated answer, and the LLM is required to select the correct one. This design enables a systematic and comparable evaluation of robustness to hallucinated adversarial examples. Therefore, we only consider zero-shot learning as the adversarial query:

$$Q^{\mathsf{hl}}_{zero} := \{Q^{\mathsf{A}}_{zero}\}.$$

IV. Evaluation Setup

In this section, we introduce our experimental settings and protocols. We first mainly elaborate on two ways of model evolution over time: upgrade and update (see Section II-B). An upgrade often involves substantial and fundamental changes. It usually does not depend on previous content and involves replacing the old with the new. On the other hand, an update refers to smaller-scale, minor, incremental, and gradual improvements within a single version. It often requires modifications (or improvements) based on the existing foundation and cannot be separated from the original basis. We then outline the corresponding datasets for testing different categories of LLMs (see Section IV-A). Finally, we introduce the evaluation metrics to summarize the results (see Section IV-B).

A. Datasets

Misclassification Datasets. We leverage the following widely used benchmarking description and question datasets to construct the clean or adversarial components of our adversarial queries under misclassification tasks.

- **Description Dataset.** We construct a curated attacking description dataset that contains the clean and adversarial descriptions from PromptBench [28]. Given a classification task, PromptBench assigns different seed descriptions, surrogate models (e.g., T5 and UL2), and adversarial attacks (e.g., TextBugger and BertAttack) to generate adversarial descriptions. Among these adversarial descriptions, we select the most prominent adversarial description according to the attack capability of the surrogate model. Then, we add the chosen clean and adversarial pairs to our curated dataset. To this end, as there are seven different categories of adversarial attacks (see Section S1), we finally generate 42 (adversarial or clean) descriptions for each question dataset to test GPT families, and 56 descriptions to query Llama models, respectively.
- **Question Dataset.** We choose GLUE [37] and AdvGLUE [31] as our question datasets for misclassification tasks. GLUE is a collection of benign classification questions for training, evaluating, and analyzing natural language understanding systems, and AdvGLUE, whose partial samples are the adversarial variants built upon GLUE samples, aims at crafting challenging and deceptive examples to evaluate the robustness of language models (see Section S2 for more details). We choose the following five benign classification tasks from GLUE as our clean question datasets: SST-2 [38] (sentiment analysis), QQP [39] (duplicate sentence detection), MNLI [40] (natural language inference), QNLI [37] (natural language inference), and RTE [37] (natural language inference), and select their corresponding adversarial versions from AdvGLUE, i.e., AdvSST-2, AdvQQP, AdvMNLI, AdvQNLI, and AdvRTE, as our final adversarial question dataset.

Jailbreak Datasets. We leverage three mainstream jailbreak algorithms, i.e., GPTfuzz [33], PAIR [35], and TAP [36], to construct our jailbreak datasets upon the samples from the Forbidden Question (FQ) dataset [20]. We aim to evaluate the effectiveness of the above three jailbreak algorithms against continuously upgraded and updated LLMs on the FQ dataset. Therefore, compared with previous works [32], the FQ dataset includes a wider variety of forbidden questions.

Hallucination Datasets. We generate the hallucination answers following the principle from [41] under three tasks, i.e., HotpotQA [42] (QA), OpenDialKG [43] (dialogue), and CNN/Daily Mail [44] (summarization). We use `mistral-7B` as a surrogate model to generate three responses for sampling 5,000 queries, with the final selection based on the lowest similarity to the correct answer, and human annotation is employed to select the hallucination answers. We construct the hallucination dataset with both the correct answer from the original dataset and the corresponding hallucinated answer for each sample.

B. Evaluation Metrics

In this paper, we consider the following three evaluation metrics (*CTS*, *RTS*, and *PDR*) for measuring the performance:

- ***Clean Test Score (CTS)*** evaluates the target LLMs on clean queries (e.g., Q^{CC}_{zero} or Q^{CCC}_{few}). Note that for different tasks, *CTS* has different meanings: (1) For the misclassification task, *CTS* signifies the *clean prediction accuracy*, thereby reflecting the normal utility of the LLMs. (2) Regarding the jailbreak task,CTS represents *rejection rate*, reflecting the capability of LLMs rejecting jailbreak attempts. Note that for the hallucination task, we do not need *CTS* to qualify the answers. Therefore, for all tasks, a *higher CTS* (↑) indicates a stronger model foundation utility.
- ***Robust Test Score (RTS)*** measures the success score given the adversarial queries, It offers multiple angles to evaluate the robustness of the LLMs on different tasks: (1) For misclassification tasks, *RTS* quantifies the prediction accuracy on adversarial examples. (2) For jailbreak tasks, *RTS* denotes the *rejection rate* of LLMs against jailbreak prompts. (3) For hallucination tasks, *RTS* stands for the correct rate for selecting the correct answers. As a result, a *higher RTS* (↑) indicates better robustness against adversarial queries.
- ***Performance Drop Rate (PDR)***, introduced by [28], aims to quantify the extent of performance decline under adversarial attacks. In general, *lower PDR* (↓) indicates superior model robustness. *PDR* can be formulated as:

$$PDR = \frac{CTS - RTS}{CTS}.$$

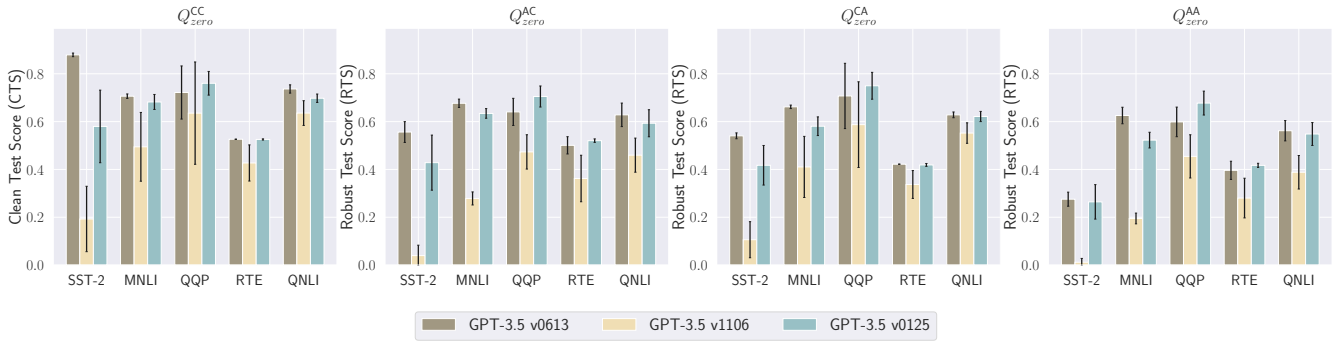


Fig. 3: *CTS* (↑) and *RTS* (↑) on GPT-3.5 under zero-shot learning.

Note. To evaluate *CTS* and *RTS* for jailbreak attacks, we use GPT-4 as our judge model. For instance, we utilized the prompt from [20], which uses three demonstrations for few-shot learning to instruct GPT-4 better to judge the harmfulness of the responses (see Figure S21).

C. Hyperparameters

In this section, we introduce the hyperparameters of our target models. For GPT models, we set the temperature to 0. For Llama and Qwen models (both loaded from the HuggingFace library), we set the do_sample to True, temperature to 0.6, and top_p to 0.9, which are the default settings of the library. For the max token, we set it to the maximum token count across the labels for each task. For misclassification and hallucination, the number is one because we increase the logistic bias in the corresponding labels. For jailbreak, we set the maximum token to 512. In the experiments, we run each task three times and report the mean and standard deviation.

V. Evaluation on GPT Families

In this section, we present our experimental results on GPT families. We outline our results in three dimensions, namely misclassification, jailbreak, and hallucination. Finally, we will show the results of the update over time.

A. Misclassification

GPT-3.5. Figure 3 and Figure 4 show the *CTS* and *RTS* results for zero-shot and few-shot learning, respectively. The *PDR* results for both are presented in Table S1 (see supplementary material). The standard deviation in the figures reflects different adversarial examples generated from various surrogate models. Results in Figure 3 indicate that v1106 has the lowest *CTS* and *RTS* among the GPT-3.5 versions and performs significantly worse across all five classification datasets in zero-shot learning. For example, v1106 *CTS* on SST-2 is 0.189, compared to 0.874 and 0.742 for the other versions. Similarly, v1106 *RTS* on Q^{AC}_{zero} is 0.038, much lower than the other models (0.556 and 0.430). Moreover, Table S1 shows v1106 has the highest *PDR* in 14 out of 15 cases, indicating the most significant performance decline. Thus, v1106 is the weakest among these versions.

Although improvements were expected in successive versions, results do not fully support this. Specifically, v0125 underperforms v0613 on many datasets, such as RTE and QNLI, with unstable behavior on adversarial examples and higher standard deviations. Despite lower *PDR* in some cases, decreased *CTS* and *RTS*, such as in SST-2, show that the upgrades do not lead to much improvement.

Figure 4 displays few-shot learning results, where the context-dependent nature of *RTS* is evident. Despite demonstrations, v1106 still performs worst in *CTS* and *RTS*. Additionally, v0125 shows no substantial improvement over v0613, with lower *RTS* in Q^{CCA}_{few} on MNLI. Table S1 highlights fluctuating *PDR* results across 28 adversarial attacks, with v0613 outperforming v0125 in 16 of them. For QQP, v0125 has a higher *PDR* despite having a larger *CTS*. This aligns with previous findings that *PDR* changes with *CTS* and *RTS*, so it cannot be the sole metric for evaluation.

Regarding the standard deviation in v1106, we inspect the query outputs and find that, for the same questions, v1106 often does not follow the predefined response template, which means a worse *instruction following ability*. Since the instruction-following behavior of v1106 is itself unstable across runs, a single evaluation would be insufficient and potentially misleading. Conducting multiple trials allows us to capture this variability and report a more faithful characterization of the model's over-time robustness.

Overall, the analysis emphasizes that specific scenarios retain significant attack effectiveness even in the upgraded version.

GPT-4. Figure S1, Figure S2, and Table S2 in the supplementary material show the robustness test of the GPT-4 family. For zero-shot learning, from Figure S1, the latest version, v0409, clearly performs the worst across the majority of datasets. In addition, only the QNLI dataset has a better *CTS* and *RTS* in the v1106 model, whereas others are comparably better on v0613. For *PDR*, an analysis of 15 adversarial attacks reveals that v0613 exhibits the lowest performance in 10 instances, while v0405 has the highest *PDR* in 10 examples. Specifically, for the MNLI and QQP datasets, v0613 consistently registers the lowest *PDR* across all evaluated settings.

Furthermore, we can see that GPT-4 demonstrates better stability for most datasets in few-shot learning on both *CTS* and *RTS* in Figure S2. However, v0409 is still the worst one. For instance, the *CTS* values of the MNLI dataset among four LLMs are 0.871, 0.837, 0.859, and 0.685, whereas the *RTS* results of Q^{ACC}_{few} are 0.877, 0.827, 0.850, and 0.652, respec-

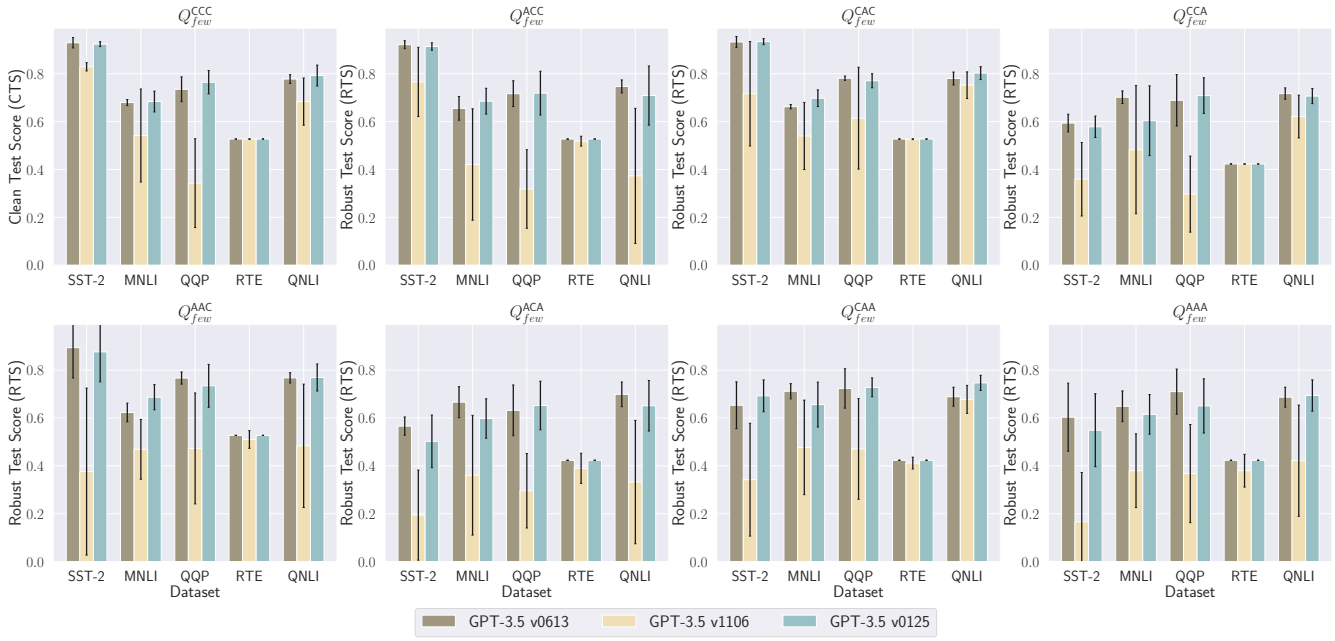


Fig. 4: *CTS* (↑) and *RTS* (↑) on GPT-3.5 under few-shot learning.

tively. For *PDR*, the upgraded versions still have the majority of the highest performance drops. For example, in the SST-2 and QNLI datasets, v0125 and v0409 consistently achieve the highest *PDR* results across all categories of adversarial attacks.

GPT-4o. Figure S3, Figure S4, and Table S3 shows the results of GPT-4o family. For zero-shot learning, as the model versions are upgraded, these adversarial examples remain effective. Meanwhile, across different datasets, the upgraded versions (v1120) do not exhibit better defense capabilities. For example, on the QNLI dataset, under Q^{AC}_{zero}, the accuracies are 0.889, 0.881, and 0.830, respectively. Moreover, in *PDR*, we can observe that, except for RTE, the results of the other datasets do not decrease with the upgrade.

For few-shot learning, we find that the latest version of GPT-4o performs worse than previous versions on many tasks. For instance, under Q^{CAA}_{few}, all v1120 models achieve worse results than the previous versions. From the perspective of PDR, the values of *PDR* vary significantly across different datasets. Although on SST-2 almost all v1120 models achieve the lowest *PDR*, the opposite trend is observed on MNLI.

Overall, the upgraded GPT-4o models do not outperform the previous versions on misclassification tasks. Therefore, we believe that such adversarial examples were not taken into account during the model upgrade process.

B. Jailbreak

GPT-3.5. The first three columns of Table S4 show that v1106 has the highest *CTS* and *RTS* among the evaluated LLMs, indicating strong resistance to jailbreak attacks. In contrast, v0613 and v0125 exhibit weaker performance in resisting various jailbreak attacks. Regarding *PDR*, as seen in Table S5 (see supplementary material), v1106 has the lowest drop rate in GPTFuzz and PAIR, while v0613 shows the lowest in TAP. This indicates that the latest version is not necessarily the most effective. Interestingly, this observation contradicts earlier findings in Section V-B, where v1106 was the least robust against misclassification.

To explore the cause of this performance discrepancy, we reviewed the introductory documents for v1106.[2] According to these documents, OpenAI introduced new systems in v1106 to ensure GPTs adhere to usage policies, building on existing mitigations to prevent harmful content such as fraud, hate, or adult themes. However, OpenAI did not elaborate on the impact of these systems on different tasks. Given the closed-source nature of these models, we hypothesize that these safety systems may contribute to the performance degradation observed in other tasks. This reveals a trade-off between misclassification performance and jailbreak resistance across all models. This trade-off presents a significant challenge for model providers, balancing optimizing LLM performance while ensuring sufficient defenses against jailbreak attempts. Recognizing this trade-off highlights the complexity and nuanced balance required in LLMs to address diverse safety and performance goals.

GPT-4. As shown in Table S4, v1106 exhibits the highest *CTS*, while v0613, with a *CTS* of 0.537, shows a significant vulnerability to jailbreak attacks, posing a substantial safety risk. For all three attacks, v0125 has the highest *RTS*, demonstrating its effectiveness in mitigating these threats, while v0409 has the lowest *RTS* for GPTFuzz. However, v0613 performs the worst in the other two attacks. Since v0613 is the default version of GPT-4, users who do not manually select a version may be more exposed to security threats. For *PDR*, as shown in Table S5 (see supplementary material), although v0613 and v0409 show lower results in PAIR and TAP, these are primarily due to lower *CTS* and *RTS*. Model developers must take a holistic approach to safety and security, incorporating jailbreak resistance during updates to enhance LLM robustness against

[2]https://openai.com/index/introducing-gpts/

various vulnerabilities.

GPT-4o. From Table S4, the upgraded GPT-4o model (v1120) indeed defends adversarial jailbreak attacks better than the previous versions. The only difference is that, for PAIR, v0806 achieves a higher *PDR*, as demonstrated in Table S5, indicating that the score decreases less under optimized adversarial prompts. In GPT-4o, we believe that as jailbreak attacks became one of the most trendy topics in LLM safety in 2024, it was inevitable for OpenAI to optimize against them; therefore, this result is within expectations.

C. Hallucination

GPT-3.5. As shown in the first three columns of Table S6, the upgrade of GPT-3.5 does not consistently reduce hallucination incidents. Specifically, v0613 has the highest *RTS* in the Dialogue and QA datasets, demonstrating better robustness in these areas but the lowest in the Summarization dataset. Conversely, v1106 has the lowest *RTS* in Dialogue and QA, while v0125 is the most robust in Summarization.

GPT-4. GPT-4 generally shows better proficiency in hallucination tests. As indicated in Table S6, v0125 achieves the highest *RTS* in the Dialogue and QA datasets, while v1106 performs best in Summarization. However, the latest version, v0409, shows a significant performance regression compared to v0125, indicating that it is not the best option for mitigating hallucination.

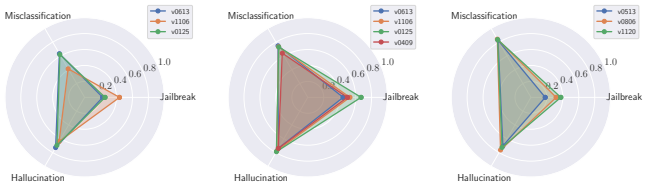
GPT-4o. From Table S6, we can see that, for dialogue and QA tasks, with the upgrade, GPT-4o models are improved against hallucination. However, for the summarization task, the *RTS* of v1120 is the worst. These results highlight the complexity of addressing hallucinations and emphasize the need for targeted multifaceted improvements.

D. Update Over Time

These longitudinally updated models use continuous self-optimization driven by user input and feedback. Regular exposure to new data and evolving use cases necessitates weekly testing to gauge their adaptability and learning efficacy. Given time limitations, we could not conduct all the adversarial attacks outlined above for each model weekly. Thus, the misclassification task was chosen to assess robustness and monitor the trajectory of model updates.

GPT-3.5. Figure S5 and Figure S6 (see supplementary material) depict the weekly test trajectory in the benchmark dataset. A significant turning point for zero-shot learning emerges between February 12th and 19th, 2023. In particular, SST-2, MNLI, and QNLI datasets show a notable decline in *CTS*, coupled with a corresponding decrease in *RTS* to varying degrees. In contrast, the QQP dataset exhibits an upward trend in both *CTS* and *RTS* during this period. For the RTE dataset, *CTS* remains relatively stable, with no major fluctuations, but *RTS* increases. Post-update to v0125, minor fluctuations persist. It is hypothesized that OpenAI may continue minor updates, potentially causing variations in attack performance.

Furthermore, Figure S6 reveals another turning point from February 12th to 19th, 2023. Compared to zero-shot learning, few-shot learning shows relatively stable performance across



(a) GPT-3.5 **(b)** GPT-4 **(c)** GPT-4o

Fig. 5: Performance of different GPT models on three tasks. We average the results of each method per task for comparison.

datasets, with minimal metric fluctuations. Notably, the QQP dataset is the only one showing a significant upward trend in *CTS*. Furthermore, the impacts of various elements of ICL on different datasets are heterogeneous, with some datasets, such as MNLI in Q_{few}^{AAC}, experiencing increases in vulnerability while others, such as SST-2 in Q_{few}^{ACA}, decrease.

By reviewing the official release notes, we find that on February 16th, `gpt-3.5-turbo` automatically updated from `gpt-3.5-turbo-0613` to `gpt-3.5-turbo-0125`. Moreover, models at slots have not seen significant updates. However, based on our experiments, this update has partially contributed to the improvements in the GPT-3.5 model, marking a clear divergence in performance metrics from its predecessors. We hope OpenAI will continue to evaluate overall performance when rolling out new features to maintain stability and consistency.

GPT-4. Compared to GPT-3.5, GPT-4 displays minimal fluctuations in both *CTS* and *RTS* during zero-shot and few-shot learning (see Figure S7 and Figure S8 in supplementary material). This indicates that, for the GPT-4 model, minor updates have not had a significant impact on its performance.

E. Takeaway

In general, although OpenAI performs optimizations for specific tasks during the GPT model upgrade process, the overall robustness does not improve with longitudinal version changes. We use Figure 5 to illustrate model comparisons across different tasks, highlighting the improvements and weaknesses between versions. Through our study, we hope to encourage OpenAI to pursue a more comprehensive optimization strategy in future upgrades.

VI. Evaluation on Llama Families

In this section, we will demonstrate our experimental results of longitudinal versions of Llama models.

A. Misclassification

Llama-7B. We show our results in Figure S9, Figure S10 and Table S7. For zero-shot learning, upon analysis, we observe that the latest Llama versions do not consistently improve across several datasets. For instance, in QNLI, Llama models capture the intended meaning but do not exactly match the desired labels. In the PromptBench QNLI dataset, "not_entailment" is split into five tokens by the tokenizer, introducing complexity into the evaluation process. Furthermore, the upgraded versions do not achieve the lowest *PDR* in most

cases, indicating that they are less resistant to adversarial attacks than the v1 models. We further visualize QQP examples in Table S10 and Table S11.

For few-shot learning, we observe results similar to those of zero-shot learning. In addition, it is still noteworthy that the *PDR* metric is a quotient derived from the division of *CTS* and *RTS*. Considering the aforementioned analysis, it becomes apparent that the smaller *PDR* values can be attributed to potential reductions in both *CTS* and *RTS*.

Llama-13B. We show the results of different Llama-13B versions in Figure S11, Figure S12 and Table S8. For zero-shot learning, when launching adversarial attacks, all upgraded models exhibit *RTS* values lower than those of the previous versions, regardless of whether they are influenced by label tokens. In this scenario, although *PDR* improves with the upgrade, we believe that Llama-13B is not optimized for adversarial examples during its upgrade.

For zero-shot learning, although the upgraded model shows a clear improvement on the SST-2 dataset, the influence of labels persists, as reflected by the fact that *PDR* remains unchanged on some datasets. We argue that this is related to the model utility. Under fair testing conditions, we do not discuss model utility; however, we believe that the continued optimization of Llama has not resulted in further improvements in robustness.

Llama-70B. We present the results in Figure S13, Figure S14, and Table S9. For zero-shot learning, the upgraded versions (especially v3I) are not as good as we expect. For the QNLI dataset, the v3I model performs the worst in most cases, and this is due to the labels again, although the *PDR* is the best in these cases.

For few-shot learning, the v3I model also does not perform well in tests against adversarial examples, showing almost no improvement in nearly all cases.

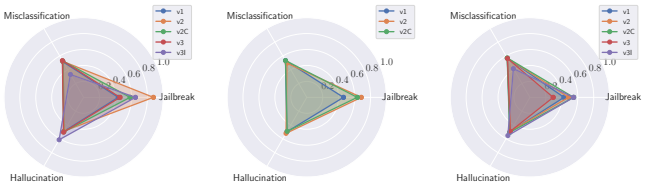
B. Jailbreak

We first show the results of adversarial-based jailbreak attacks in Table S12 and Table S13.

Llama-7B. From the tables, *CTS* values across all versions approach 1.000, indicating that clean queries are ineffective against these models. For *RTS*, the v2 model consistently shows the highest value across all attacks. However, the v3 model has the lowest *RTS* in GPTFuzz and TAP and the second-worst in PAIR. The *RTS* of v3I for GPTFuzz is lower than v2C, but it performs better in the other two attacks. For *PDR*, the v3 model shows the highest drop rate for both GPTFuzz and TAP.

Llama-13B. For the 13B family, *CTS* values are near 1.000, signaling strong defenses against jailbreak prompts. The v1 model has a higher *RTS* and lower *PDR* in GPTFuzz than later versions, but v2 shows an increasing *RTS* for the other two attacks, with v2 Chat following a similar trend. As shown in Table S13, v2 models also have lower *PDR* results in PAIR and TAP.

Llama-70B. In the 70B family, *CTS* results are similar to the 7B and 13B models, close to 1.000. For GPTFuzz, the v1 model is more resistant to attacks, with higher *RTS* and lower *PDR* compared to upgraded versions. In the other two attacks, v2C shows better resilience, while the v3 model has the highest *RTS* and lowest *PDR* across all attacks.



(a) Llama 7B **(b)** Llama 13B **(c)** Llama 70B

Fig. 6: Performance of different Llama models on three tasks. We average the results of each method per task for comparison.

Transverse Comparison. Prior work [45] suggested that larger models, like the 70B, are safer than smaller ones, like the 13B. However, our results do not support this conclusion, especially for v2 and v3 models. Larger models, despite increased computational capacity and more sophisticated learning, consistently show lower *RTS* compared to smaller ones, as shown in Table S12. This suggests that larger models may offer a larger attack surface, making them more vulnerable to these attacks. Increasing model size does not straightforwardly enhance security. Instead, it may introduce new vulnerabilities or amplify existing ones.

C. Hallucination

Table S14 lists all the results of hallucination task for Llama families.

Llama-7B. For the 7B family, the latest version v3I is the best among the three datasets. However, some results are still around 0.500, such as summarization. This raises concerns about the current training methods for handling complex tasks prone to hallucination, underscoring the need for model providers to incorporate broader foundational knowledge and adopt advanced training approaches in future upgrades.

Llama-13B. Similar to the Llama-7B family, the v2 and v2C models perform better than v1, but their results on hallucination prompts remain nearly random.

Llama-70B. In the 70B family, although the results of dialogue and QA datasets are the best in the upgraded version, the summarization datasets still perform worse. We also hope that Meta will incorporate hallucination into the safety tasks in future model upgrades.

D. Takeaway

In a nutshell, we believe that robustness was not sufficiently considered during the upgrade process. In many tests, the latest model versions perform worse, and larger models exhibit inferior performance on certain tasks. We hope that Meta will take these issues into greater account in future upgrades. Based on these models, we also present radar figures in Figure 6 to facilitate better comparisons across versions in future research.

VII. Evaluation on Qwen Families

In this section, we present the results for the Qwen families.

A. Misclassification

Qwen-7B. Figure S15, Figure S16, and Table S15 demonstrate the reuslts of Qwen-7B family. For zero-shot learning, the robustness of the upgraded model (v3) is not better than that of the previous versions. Due to PromptBench, some datasets still cannot obtain the exact label from the outputs of Qwen models. For the *PDR*, the latest versions of the 7B family are still worse than the previous version.

For few-shot learning, the results are similar to those of zero-shot learning.

Qwen-32B. We present the results of Qwen-32B family in Figure S17, Figure S18, and Table S16. For zero-shot learning, in *CTS*, the upgraded versions do not demonstrate better performance. Moreover, when facing adversarial examples, the upgraded models exhibit an overall downward trend. This behavior is also reflected in *PDR*: among the 15 results, 11 upgraded models have worse values than their previous versions. This indicates that Qwen did not place sufficient emphasis on model robustness during the update process.

For few-shot learning, the latest version performs worse than previous models, particularly on the MNLI and QQP datasets. From the *PDR* results, the v3 model is not the best among all the versions. For example, in the SST-2 dataset, the v3 model is the worst, demonstrating that the upgrade will not eliminate its weakness to adversarial examples.

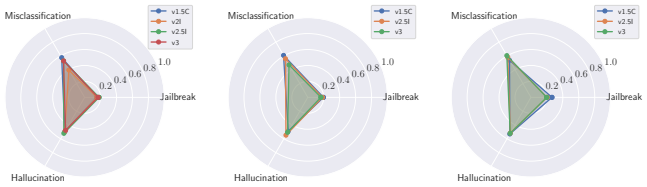
Qwen-72B. Figure S19, Figure S20, and Table S17 show the results of Qwen-72B family against misclassification. For zero-shot learning, one apparent contradiction is that, in *CTS* and *RTS*, upgrades do not improve robustness on datasets such as SST-2 and MNLI, whereas from the perspective of *PDR*, upgrades lead to lower values. That is, when considering the ratio and test accuracy alone, these two observations appear contradictory. Therefore, from a more comprehensive perspective, the upgraded models are insufficient to demonstrate an improvement in robustness. More in-depth research is needed to provide a comprehensive definition.

For zero-shot learning, we observe a similar result to the zero-shot learning. In addition, we emphasize that in the Qwen models, especially under few-shot learning, adversarial questions have a greater impact on overall robustness, leading to larger drops in *RTS*. We analyze all Qwen families and find that in the 7B family, 12 out of 20 models exhibit lower *RTS* under adversarial questions than under benign questions; in the 32B family, this holds for 9 out of 15 models; and in the 72B family, for 8 out of 15 models—each exceeding half of the models in the respective family.

B. Jailbreak

We first show our jailbreak results in Table S18 and Table S19.

Qwen-7B. From the table, we know that for the 7B family, the upgraded models exhibit greater resistance to jailbreak attacks in both *CTS* and *RTS*. Although from the perspective of *PDR*, the upgraded models have the highest values, this is due to their very high *CTS*, which makes the ratio large. This implies that, when considering only the relative drop, the upgraded models are more susceptible to adversarial-attack optimization; however, their overall attack success rate remains lower than that of the previous models.



(a) Qwen 7B **(b)** Qwen 32B **(c)** Qwen 72B

Fig. 7: Performance of different Qwen models on three tasks. We average the results of each method per task for comparison.

Qwen-32B. In the 32B family, the latest version is not the best-performing one. Although the v3 model achieves the highest value in *CTS*, its *RTS* is worse than that of previous models when facing adversarial-based optimization, which is also reflected in *PDR*. Therefore, we believe that during the update process of the 32B models, the developers did not place sufficient emphasis on safety alignment.

Qwen-72B. For *CTS*, the 72B family, similar to the previous two families, achieves very high values, indicating that the optimization against jailbreak-in-the-wild is indeed effective. However, when facing adversarial examples, the upgrade does not make the model safer. Therefore, we believe that Qwen models should further improve jailbreak robustness in future upgrades to better defend against such optimization-based adversarial examples.

C. Hallucination

Table S20 demonstrates the results of the longitudinal versions of Qwen families.

Qwen-7B. For the 7B version, the upgraded model does not perform best among all the models. For the dialogue and QA task, the v3 model performs the worst. This means that the upgrade process is not optimized against hallucinations.

Qwen-32B. Among the 32B models, the best version is v2.5I, not the latest version (v3).

Qwen-72B. In this scenario, the upgraded model can perform best in the summarization task, while performing worst in dialogue tasks. In addition, these *RTS* values are all around 0.500. This indicates that the gap between the best and the worst is small, and the overall performance of the model family is poor.

D. Takeaway

Overall, we believe that robustness was not sufficiently considered during the upgrade process of the Qwen models. As a result, when facing adversarial examples, model utility degrades rapidly. We argue that future model updates should place greater emphasis on strengthening robustness. We also present radar figures in Figure 7 for different model families across various tasks, to facilitate future research in drawing more robust conclusions about their robustness.

VIII. Possible Explanation

Here, we take the initiative to investigate the underlying causes of our observations empirically. Recall that we measure

TABLE V: Performance of Qwen-7B 2I original and safety fine-tuned model against jailbreak attacks.

Jailbreak	Qwen-7B 2I	
	Origin	Fine-tuned
GPTFuzz	0.137	0.438
PAIR	0.169	0.863
TAP	0.156	0.906
CTS	0.312	0.812

TABLE VI: Performance of Qwen-7B 2I original and safety fine-tuned model against misclassification.

Misclassification		SST-2	MNLI	QQP	RTE	QNLI
Zero-shot	**Origin**	0.838	0.577	0.524	0.019	0.053
	Fine-tuned	0.522	0.383	0.382	0.172	0.168
Few-shot	**Origin**	0.806	0.588	0.501	0.022	0.055
	Fine-tuned	0.524	0.383	0.382	0.116	0.066

mainstream LLMs in this paper. However, due to the lack of access to their proprietary training processes and datasets of these LLMs, we employed Qwen2 [7] as representative transformer models for our analysis. We finetune the model by using LLaMA-Factory [46] with the STAIR-SFT [47] dataset. This dataset is built for the safety alignment of LLMs, containing about 20,000 samples.

We first present the jailbreak attack results for the original Qwen-7B 2I model and the fine-tuned model in Table V. From the table, we see that after fine-tuning, both *CTS* and *RTS* perform much better than before on all tasks, indicating that model providers can improve model safety by fine-tuning.

On the other hand, Table VI demonstrates the results of original and fine-tuned Qwen-7B 2I models. From the table, we observe that after fine-tuning, the model performs worse on the SST-2, QNLI, and QQP datasets. This is very similar to the models in our main experiments. This implies that optimizing for a single task alone can lead to unstable effects on the model's overall robustness. In particular, as safety alignment–related studies are increasingly becoming a dominant topic, focusing solely on safety alignment may cause unpredictable consequences for robustness on other tasks. Therefore, we hope LLM providers to place greater emphasis on overall model robustness in future upgrades and updates, rather than optimizing for a single task.

IX. Related Work

Adversarial Attacks. We focus on adversarial attacks that manipulate legitimate inputs to mislead a trained model to produce incorrect outputs in the NLP domain [48]. These attacks commonly manipulate the input text at character-, word-, and sentence-level to attain the attack goals (i.e., targeted or untargeted attacks). Similar to adversarial attacks in computer vision domain, they can be categorized into black-box attacks (paraphrase [49]–[51], text manipulation [52]–[54], etc.) and white-box attacks (FGSM [55], [56], JSMA [57], HotFlip [58], etc.). In the NLP domain, those attacks have been successfully applied to attack various applications, such as optical character recognition [59], image caption [60], visual question answering [61], etc. Our objective here is not to devise novel adversarial attacks against LLMs. Rather, we use existing methods to understand whether LLMs can be challenged by carefully crafted textual adversarial examples and whether/how these adversarial examples can be transferred to different versions of an LLM.

Jailbreak. Previous works have explored multiple paradigms for obtaining effective jailbreak prompts. These include harvesting prompts from real-world user interactions and deployments [62], designing prompts through human-crafted and strategy-guided constructions [63], [64], as well as leveraging automated frameworks to synthesize jailbreak inputs at scale [32]–[36]. Beyond prompt design, it has also been shown that the alignment mechanisms of LLMs are inherently incomplete: even without modifying the input query, certain configurations of generation parameters can still induce the model to produce harmful outputs [65]. In addition, some studies [20], [66] systematically evaluate diverse jailbreak techniques within a unified experimental framework. They typically benchmark multiple jailbreak strategies across a wide range of models and safety settings, and further consolidate the resulting prompts and responses into standardized evaluation datasets.

Hallucination. Many studies on hallucination analysis in LLMs investigate the phenomenon by leveraging the models themselves as analytical instruments. When access to internal representations is available, prior work has examined model internals to uncover mechanisms underlying hallucinated generations [16], [67], [68]. These studies commonly analyze signals derived from output logits, intermediate hidden representations, or attention patterns. Yu et al. [69] illustrated that nonsense prompts, consisting of random tokens, can prompt LLMs to generate hallucinations, indicating that hallucinations might be viewed as another form of adversarial examples. Li et al. [41] explored the perceptual capabilities of LLMs concerning the boundaries of factual knowledge.

LLMs. Large language models (LLMs) have become a prominent area of research and application in the NLP domain, driven primarily by the transformer architecture [70]. These models are trained on massive text data and boast a substantial number of parameters, often exceeding hundreds of billions [71]. As LLMs grow in size, they demonstrate emergent abilities such as enhanced language understanding [72], coherent text generation [73], and contextual comprehension [74], which are not present in smaller models. Moreover, fine-tuning techniques (e.g., LoRA [75]) are invented to adapt the pre-trained LLMs to specific downstream tasks, allowing them to exhibit specialized behavior and produce task-specific outputs.

X. Discussion

Discussions. Robustness is essential for AI systems, as required by the EU AI Act.[3] However, our findings indicate that neither open-source nor closed-source LLMs exhibit consistent improvements in robustness over time, challenging the assumption that model upgrades inherently lead to increased reliability. This suggests that robustness should be treated as an independent and continuously evaluated property rather than an implicit outcome of model scaling or iteration. Looking forward, our results motivate the need for lightweight yet systematic robustness evaluations to be integrated into the

[3]https://artificialintelligenceact.eu/

LLM update lifecycle, enabling the detection of robustness regressions across versions. In addition, incorporating adversarial perturbations into training or alignment procedures may help decouple generative quality from robustness under misleading inputs. Improved transparency in release documentation—particularly regarding training data changes, alignment strategies, and robustness self-assessments—would allow practitioners to better understand and manage the robustness implications of model upgrades and updates. Finally, we will examine whether robustness gains from LLM upgrades are attack-class specific, for example, prioritizing practical, human-written, or low-effort attacks over optimized or machine-generated ones [76]. In addition, analyzing which properties of jailbreaking attacks [77] are explicitly addressed—or remain unaddressed—by model updates may help distinguish attack-specific fixes from general robustness improvements across the LLM lifecycle.

Limitations. Despite yielding valuable insights, our study has several limitations. First, we did not generate adversarial examples for the *Adversarial Description* and *Adversarial Question* datasets, opting instead for existing datasets from [78], primarily due to the significant cost of querying GPT models for several weeks. Second, there is no universal template for ICL, as some queries are only effective for specific datasets, and minor changes in wording can drastically alter classification outcomes, complicating the query process. Additionally, evaluating LLM outputs remains an open question, as no single method is optimal for all tasks. Lastly, since OpenAI and Meta have not open-sourced their training datasets, there is a potential risk of inadvertently testing models on data they were trained on. Our analysis indicates a low probability of overlap between our evaluation dataset and their training data, though accurately assessing this for future models remains a challenge. It is increasingly important to construct or select non-overlapping datasets for LLM evaluation, a trend we may follow by using CC BY-SA 4.0 licensed or custom datasets.

XI. Conclusion

We comprehensively assess the robustness of the longitudinal versions of LLMs, focusing on GPT, Llama, and Qwen families through the lens of misclassification, jailbreak, and hallucination evaluation. Our empirical results consistently demonstrate that, for all the LLMs, the upgraded and updated model does not exhibit heightened robustness against the proposed adversarial queries compared to its predecessor. In addition, an increase in model size does not guarantee improved robustness, especially for Llama families. Qwen models are more vulnerable to adversarial questions than other content. Our findings reinforce the importance of understanding and assessing the robustness aspect when upgrading and updating LLMs, calling for enhanced focus on comprehensive evaluation and reinforcement strategies to counter evolving adversarial challenges.

References

[1] https://platform.openai.com/docs/models/gpt-3-5-turbo.
[2] OpenAI, "GPT-4o," https://openai.com/index/hello-gpt-4o/.
[3] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and Efficient Foundation Language Models," *CoRR abs/2302.13971*, 2023.
[4] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Canton-Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open Foundation and Fine-Tuned Chat Models," *CoRR abs/2307.09288*, 2023.
[5] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Rozière, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. M. Kloumann, I. Misra, I. Evtimov, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, and et al., "The Llama 3 Herd of Models," *CoRR abs/2407.21783*, 2024.
[6] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu, "Qwen2.5 Technical Report," *CoRR abs/2412.15115*, 2024.
[7] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, and Z. Fan, "Qwen2 Technical Report," *CoRR abs/2407.10671*, 2024.
[8] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu, "Qwen3 Technical Report," *CoRR abs/2505.09388*, 2025.
[9] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, B. Hui, L. Ji, M. Li, J. Lin, R. Lin, D. Liu, G. Liu, C. Lu, K. Lu, J. Ma, R. Men, X. Ren, X. Ren, C. Tan, S. Tan, J. Tu, P. Wang, S. Wang, W. Wang, S. Wu, B. Xu, J. Xu, A. Yang, H. Yang, J. Yang, S. Yang, Y. Yao, B. Yu, H. Yuan, Z. Yuan, J. Zhang, X. Zhang, Y. Zhang, Z. Zhang, C. Zhou, J. Zhou, X. Zhou, and T. Zhu, "Qwen Technical Report," *CoRR abs/2309.16609*, 2023.
[10] W. Jiao, W. Wang, J. Huang, X. Wang, and Z. Tu, "Is ChatGPT A Good Translator? A Preliminary Study," *CoRR abs/2301.08745*, 2023.
[11] X. Sun, X. Li, J. Li, F. Wu, S. Guo, T. Zhang, and G. Wang, "Text Classification via Large Language Models," *CoRR abs/2305.08377*, 2023.
[12] Y. Ji, Y. Deng, Y. Gong, Y. Peng, Q. Niu, L. Zhang, B. Ma, and X. Li, "Exploring the Impact of Instruction Data Scaling on Large Language Models: An Empirical Study on Real-World Use Cases," *CoRR abs/2303.14742*, 2023.
[13] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, B. Newman, B. Yuan, B. Yan, C. Zhang, C. Cosgrove, C. D. Manning, C. Ré, D. Acosta-Navas,

D. A. Hudson, E. Zelikman, E. Durmus, F. Ladhak, F. Rong, H. Ren, H. Yao, J. Wang, K. Santhanam, L. J. Orr, L. Zheng, M. Yüksekgönül, M. Suzgun, N. Kim, N. Guha, N. S. Chatterji, O. Khattab, P. Henderson, Q. Huang, R. Chi, S. M. Xie, S. Santurkar, S. Ganguli, T. Hashimoto, T. Icard, T. Zhang, V. Chaudhary, W. Wang, X. Li, Y. Mai, Y. Zhang, and Y. Koreeda, "Holistic Evaluation of Language Models," *CoRR abs/2211.09110*, 2022.

[14] A. Abid, M. Farooqi, and J. Zou, "Persistent Anti-Muslim Bias in Large Language Models," in *AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2021, pp. 298–306.

[15] Y. Jiang, Z. Li, X. Shen, Y. Liu, M. Backes, and Y. Zhang, "ModSCAN: Measuring Stereotypical Bias in Large Vision-Language Models from Vision and Language Modalities," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2024, pp. 12 814–12 845.

[16] A. Azaria and T. Mitchell, "The Internal State of an LLM Knows When its Lying," *CoRR abs/2304.13734*, 2023.

[17] P. Manakul, A. Liusie, and M. J. F. Gales, "SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models," *CoRR abs/2303.08896*, 2023.

[18] Y. Pan, L. Pan, W. Chen, P. Nakov, M. Kan, and W. Y. Wang, "On the Risk of Misinformation Pollution with Large Language Models," *CoRR abs/2305.13661*, 2023.

[19] H. W. A. Hanley and Z. Durumeric, "Machine-Made Media: Monitoring the Mobilization of Machine-Generated Articles on Misinformation and Mainstream News Websites," *CoRR abs/2305.09820*, 2023.

[20] J. Chu, Y. Liu, Z. Yang, X. Shen, M. Backes, and Y. Zhang, "JailbreakRadar: Comprehensive Assessment of Jailbreak Attacks Against LLMs," in *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2025, pp. 21 538–21 566.

[21] D. Kang, X. Li, I. Stoica, C. Guestrin, M. Zaharia, and T. Hashimoto, "Exploiting Programmatic Behavior of LLMs: Dual-Use Through Standard Security Attacks," *CoRR abs/2302.05733*, 2023.

[22] A. Borji, "A Categorical Archive of ChatGPT Failures," *CoRR abs/2302.03494*, 2023.

[23] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2020.

[24] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, 2020.

[25] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, élio Renard Lavaud, M. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7B," *CoRR abs/2310.06825*, 2023.

[26] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata, "Zero-Shot Learning - A Comprehensive Evaluation of the Good, the Bad and the Ugly," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[27] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned Language Models are Zero-Shot Learners," in *International Conference on Learning Representations (ICLR)*, 2022.

[28] K. Zhu, J. Wang, J. Zhou, Z. Wang, H. Chen, Y. Wang, L. Yang, W. Ye, N. Z. Gong, Y. Zhang, and X. Xie, "PromptBench: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts," *CoRR abs/2306.04528*, 2023.

[29] B. Wang, W. Chen, H. Pei, C. Xie, M. Kang, C. Zhang, C. Xu, Z. Xiong, R. Dutta, R. Schaeffer, S. T. Truong, S. Arora, M. Mazeika, D. Hendrycks, Z. Lin, Y. Cheng, S. Koyejo, D. Song, and B. Li, "DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models," *CoRR abs/2306.11698*, 2023.

[30] J. Wang, X. Hu, W. Hou, H. Chen, R. Zheng, Y. Wang, L. Yang, H. Huang, W. Ye, X. Geng, B. Jiao, Y. Zhang, and X. Xie, "On the Robustness of ChatGPT: An Adversarial and Out-of-distribution Perspective," *CoRR abs/2302.12095*, 2023.

[31] B. Wang, C. Xu, S. Wang, Z. Gan, Y. Cheng, J. Gao, A. H. Awadallah, and B. Li, "Adversarial GLUE: A Multi-Task Benchmark for Robustness Evaluation of Language Models," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2021.

[32] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson, "Universal and Transferable Adversarial Attacks on Aligned Language Models," *CoRR abs/2307.15043*, 2023.

[33] J. Yu, X. Lin, Z. Yu, and X. Xing, "GPTFUZZER: Red Teaming Large Language Models with Auto-Generated Jailbreak Prompts," *CoRR abs/2309.10253*, 2023.

[34] G. Deng, Y. Liu, Y. Li, K. Wang, Y. Zhang, Z. Li, H. Wang, T. Zhang, and Y. Liu, "Jailbreaker: Automated Jailbreak Across Multiple Large Language Model Chatbots," *CoRR abs/2307.08715*, 2023.

[35] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong, "Jailbreaking Black Box Large Language Models in Twenty Queries," *CoRR abs/2310.08419*, 2023.

[36] A. Mehrotra, M. Zampetakis, P. Kassianik, B. Nelson, H. Anderson, Y. Singer, and A. Karbasi, "Tree of Attacks: Jailbreaking Black-Box LLMs Automatically," *CoRR abs/2312.02119*, 2023.

[37] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," in *International Conference on Learning Representations (ICLR)*, 2019.

[38] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2013, pp. 1631–1642.

[39] Z. Wang, W. Hamza, and R. Florian, "Bilateral Multi-Perspective Matching for Natural Language Sentences," in *International Joint Conferences on Artifical Intelligence (IJCAI)*. IJCAI, 2017, pp. 4144–4150.

[40] A. Williams, N. Nangia, and S. R. Bowman, "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. ACL, 2018, pp. 1112–1122.

[41] J. Li, X. Cheng, X. Zhao, J.-Y. Nie, and J.-R. Wen, "HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2023, pp. 6449–6464.

[42] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. W. Cohen, R. Salakhutdinov, and C. D. Manning, "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2018, pp. 2369–2380.

[43] S. Moon, P. Shah, A. Kumar, and R. Subba, "OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs," in *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2019, pp. 845–854.

[44] A. See, P. J. Liu, and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks," in *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2017, pp. 1073–1083.

[45] X. Zhao, X. Yang, T. Pang, C. Du, L. Li, Y.-X. Wang, and W. Y. Wang, "Weak-to-Strong Jailbreaking on Large Language Models," *CoRR abs/2401.17256*, 2024.

[46] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, Z. Luo, Z. Feng, and Y. Ma, "LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models," *CoRR abs/2403.13372*, 2024.

[47] Y. Zhang, S. Zhang, Y. Huang, Z. Xia, Z. Fang, X. Yang, R. Duan, D. Yan, Y. Dong, and J. Zhu, "STAIR: Improving Safety Alignment with Introspective Reasoning," *CoRR abs/2502.02384*, 2025.

[48] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li, "Adversarial Attacks on Deep-learning Models in Natural Language Processing: A Survey," *ACM Transactions on Intelligent Systems and Technology*, 2020.

[49] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer, "Adversarial Example Generation with Syntactically Controlled Paraphrase Networks," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. ACL, 2018, pp. 1875–1885.

[50] M. T. Ribeiro, S. Singh, and C. Guestrin, "Semantically Equivalent Adversarial Rules for Debugging NLP models," in *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2018, pp. 856–865.

[51] M. Alzantot, Y. Sharma, A. Elgohary, B. Ho, M. B. Srivastava, and K. Chang, "Generating Natural Language Adversarial Examples," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2018, pp. 2890–2896.

[52] Y. Belinkov and Y. Bisk, "Synthetic and Natural Noise Both Break Neural Machine Translation," in *International Conference on Learning Representations (ICLR)*, 2018.

[53] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "TextBugger: Generating Adversarial Text Against Real-world Applications," in *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2019.

[54] P. Minervini and S. Riedel, "Adversarially Regularising Neural NLI Models to Integrate Logical Background Knowledge," in *Conference on Computational Natural Language Learning (CoNLL)*. ACL, 2018, pp. 65–74.
[55] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi, "Deep Text Classification Can be Fooled," in *International Joint Conferences on Artifical Intelligence (IJCAI)*. IJCAI, 2018, pp. 4208–4215.
[56] S. Samanta and S. Mehta, "Generating Adversarial Text Samples," *European Conference on Information Retrieval*, 2018.
[57] N. Papernot, P. D. McDaniel, A. Swami, and R. E. Harang, "Crafting Adversarial Input Sequences for Recurrent Neural Networks," in *IEEE Military Communications Conference (MILCOM)*. IEEE, 2016, pp. 49–54.
[58] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "HotFlip: White-Box Adversarial Examples for Text Classification," in *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2018, pp. 31–36.
[59] N. Shazeer and M. Stern, "Adafactor: Adaptive Learning Rates with Sublinear Memory Cost," in *International Conference on Machine Learning (ICML)*. PMLR, 2018, pp. 4603–4611.
[60] H. Chen, H. Zhang, P. Chen, J. Yi, and C. Hsieh, "Attacking Visual Language Grounding with Adversarial Examples: A Case Study on Neural Image Captioning," in *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2018, pp. 2587–2597.
[61] X. Xu, X. Chen, C. Liu, A. Rohrbach, T. Darrell, and D. Song, "Fooling Vision and Language Models Despite Localization and Attention Mechanism," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 4951–4961.
[62] X. Shen, Z. Chen, M. Backes, Y. Shen, and Y. Zhang, "Do Anything Now: Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2024.
[63] Z.-X. Yong, C. Menghini, and S. H. Bach, "Low-Resource Languages Jailbreak GPT-4," *CoRR abs/2310.02446*, 2023.
[64] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How Does LLM Safety Training Fail?" *CoRR abs/2307.02483*, 2023.
[65] Y. Huang, S. Gupta, M. Xia, K. Li, and D. Chen, "Catastrophic Jailbreak of Open-source LLMs via Exploiting Generation," *CoRR abs/2310.06987*, 2023.
[66] P. Chao, E. Debenedetti, A. Robey, M. Andriushchenko, F. Croce, V. Sehwag, E. Dobriban, N. Flammarion, G. J. Pappas, F. Tramer, H. Hassani, and E. Wong, "JailbreakBench: An Open Robustness Benchmark for Jailbreaking Large Language Models," *CoRR abs/2404.01318*, 2024.
[67] N. Varshney, W. Yao, H. Zhang, J. Chen, and D. Yu, "A Stitch in Time Saves Nine: Detecting and Mitigating Hallucinations of LLMs by Validating Low-Confidence Generation," *CoRR abs/2307.03987*, 2023.
[68] M. Yuksekgonul, V. Chandrasekaran, E. Jones, S. Gunasekar, R. Naik, H. Palangi, E. Kamar, and B. Nushi, "Attention Satisfies: A Constraint-Satisfaction Lens on Factual Errors of Language Models," *CoRR abs/2309.15098*, 2023.
[69] J.-Y. Yao, K.-P. Ning, Z.-H. Liu, M.-N. Ning, Y.-Y. Liu, and L. Yuan, "LLM Lies: Hallucinations are not Bugs, but Features as Adversarial Examples," *CoRR abs/2310.01469*, 2023.
[70] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Annual Conference on Neural Information Processing Systems (NIPS)*. NIPS, 2017, pp. 5998–6008.
[71] "ChatGPT," https://chat.openai.com/chat.
[72] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models," *CoRR abs/2304.10592*, 2023.
[73] J. J. Y. Chung, E. Kamar, and S. Amershi, "Increasing Diversity While Maintaining Accuracy: Text Data Generation with Large Language Models and Human Interventions," in *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2023, pp. 575–593.
[74] W. Zhou, S. Zhang, H. Poon, and M. Chen, "Context-faithful Prompting for Large Language Models," *CoRR abs/2303.11315*, 2023.
[75] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *International Conference on Learning Representations (ICLR)*, 2022.
[76] T. Le, J. Lee, K. Yen, Y. Hu, and D. Lee, "Perturbations in the Wild: Leveraging Human-Written Text Perturbations for Realistic Adversarial Attack and Defense," in *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2022, pp. 2953–2965.
[77] Y. Zeng, H. Lin, J. Zhang, D. Yang, R. Jia, and W. Shi, "How Johnny Can Persuade LLMs to Jailbreak Them: Rethinking Persuasion to Challenge AI Safety by Humanizing LLMs," *CoRR abs/2401.06373*, 2024.
[78] K. Zhou, Y. Zhu, Z. Chen, W. Chen, W. X. Zhao, X. Chen, Y. Lin, J. Wen, and J. Han, "Don't Make Your LLM an Evaluation Benchmark Cheater," *CoRR abs/2311.01964*, 2023.

BIOGRAPHY SECTION

Yugeng Liu is a Ph.D. student at CISPA Helmholtz Center for Information Security. His main research interests include trustworthy machine learning.

Tianshuo Cong is a postdoctoral researcher at the Institute for Advanced Study, Tsinghua University (IASTU), hosted by Prof. Xiaoyun Wang (IACR Fellow). He received his Ph.D. degree from the Institute for Advanced Study, Tsinghua University in 2023.

Zhengyu Zhao is a faculty member at Xi'an Jiaotong University, China. In 2021-2023, he was a postdoc at CISPA Helmholtz Center for Information Security, Germany, hosted by Prof. Michael Backes and Dr. Yang Zhang. In 2021, he received my PhD from Radboud University, Netherlands, supervised by Prof. Martha Larson.

Michael Backes (Fellow, IEEE) is the founding director and CEO of the CISPA Helmholtz Center for Information Security. He focuses primarily on trustworthy machine learning methods, novel approaches for protecting personal data, and universal software and system security solutions. He has published over 300 peer-reviewed and highly cited publications in renowned international journals and conference proceedings. His research has earned him internationally renowned scientific awards and honors, particularly the Karl Heinz Beckurts Prize, the Caspar Bowden Privacy Award, the IEEE and ACM Fellowship, and various Career Awards and Best Paper Awards. He holds an honorary doctorate from the Université de Lorraine and is an honorary citizen and future ambassador of the city of St. Ingbert, Germany.

Yun Shen is a cybersecurity and AI researcher focusing on adversarial machine learning, data privacy, and the security and safety of large language models (LLMs) and generative models. His work regularly appears at venues like USENIX Security, IEEE S&P, CCS, and NDSS.

Yang Zhang (Member, IEEE) is a faculty member at CISPA Helmholtz Center for Information Security. His research concentrates on trustworthy machine learning. Moreover, he works on measuring and understanding misinformation and unsafe content like hateful memes on the Internet. Over the years, he has published multiple papers at top venues in computer science, including CCS, NDSS, Oakland, and USENIX Security. His work has received the NDSS 2019 Distinguished Paper Award and the CCS 2022 Best Paper Award runner-up.