HIJACKING LARGE LANGUAGE MODELS VIA ADVERSARIAL IN-CONTEXT LEARNING

by

YAO QIANG

DISSERTATION

Submitted to the Graduate School,

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

2024

MAJOR: COMPUTER SCIENCE

Approved By:

_____

Advisor Date

_____

Committee Member 1 Date

_____

Committee Member 2 Date

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. Dongxiao Zhu, for his guidance, support, and encouragement throughout the development of this work. I am especially thankful to Xiangyu Zhou, whose insights, technical help, and many discussions significantly strengthened the design of the application and the experimental framework. I also appreciate Yao Qiang for his valuable suggestions on the experiments and manuscript revisions.

My sincere thanks go to Srinivasan Suresh, Steven Hicks, and Usha Sethuraman for their contributions to data collection and for the helpful feedback that shaped several stages of this project. I am grateful to the members of the Trustworthy AI Lab for creating an environment that supported both the research and implementation aspects of this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Background

In-context learning (ICL) is an emerging technique for rapidly adapting large language models (LLMs), i.e., GPT-4 [**?**] and LLaMA2 [**?**], to new tasks without fine-tuning the pre-trained parameters [**?**]. The key idea behind ICL is to provide LLMs with labeled examples as in-context demonstrations (demos) within the prompt context before a test query. Through learning from the demos, LLMs are able to generate responses to queries based on ICL [**?**, **?**].

Several existing works, however, have demonstrated the highly unstable nature of ICL [**?**, **?**]. Specifically, performance on target tasks using ICL can vary wildly based on the selection and order of demos, giving rise to highly volatile outcomes ranging from random to near state-of-the-art (SOTA) [**?**, **?**, **?**, **?**, **?**]. Correspondingly, several approaches [**?**, **?**, **?**] have been proposed to address the unstable issue of ICL.

Further research has looked at how adversarial examples can undermine the performance of ICL [**?**, **?**, **?**, **?**]. These studies show that maliciously designed examples injected into the prompt instructions [**?**, **?**, **?**], demos [**?**, **?**], or queries [**?**, **?**] can successfully attack LLMs to degrade their performance, revealing the significant vulnerabilities of ICL against adversarial inputs.

While existing adversarial attacks have been applied to evaluate LLM robustness, they have some limitations in practice. Most character-level attacks, e.g., TextAttack [**?**] and TextBugger [**?**], can be easily detected and evaded through grammar checks, limiting real-world effectiveness [**?**, **?**]. Some other attacks like BERTAttack [**?**] require an extra model

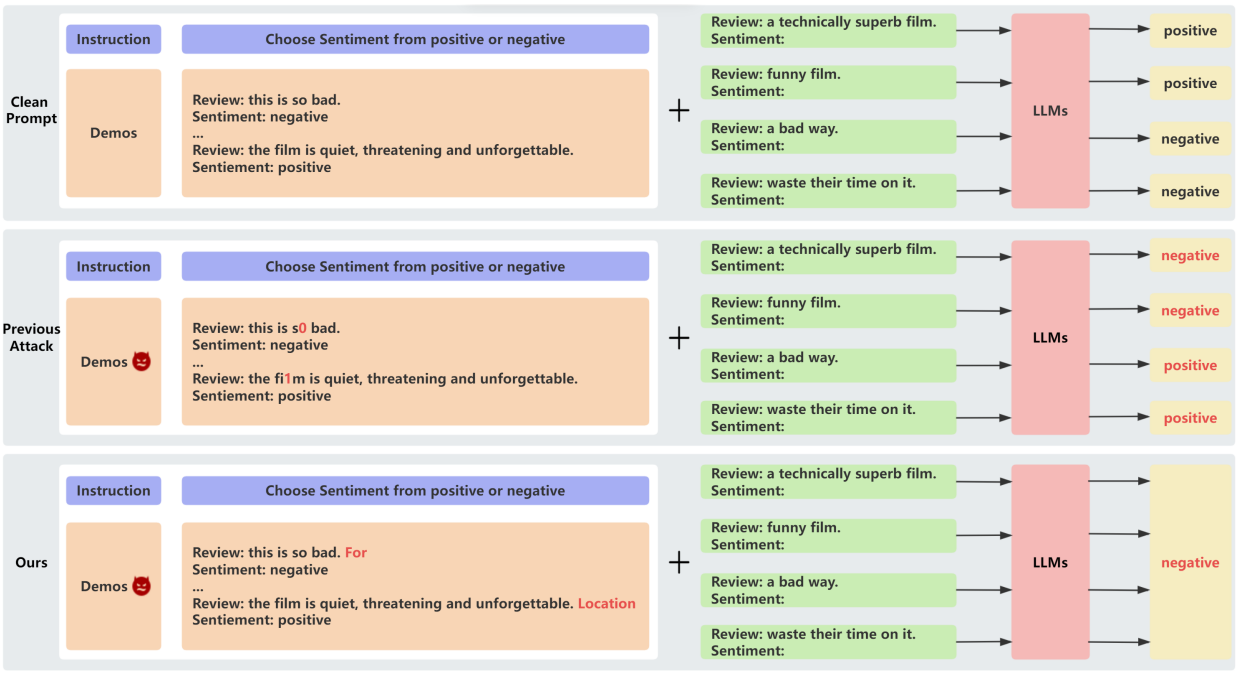


Figure 1: Illustrations of ICL using clean prompt and adversarial prompt. Given the clean in-context demos, LLMs can correctly generate the sentiment of the test queries. The previous attacks [**?**] at the character level involve minor edits in some words, such as altering 'so' to 's0' and 'film' to 'fi1m', of these in-context demos, leading to incorrect sentiment generated for the test queries. However, ours **learns** to **append** adversarial suffixes like 'For' and 'Location' to the in-context demos to efficiently and effectively **hijack** LLMs to generate the **unwanted target**, e.g., the 'negative' sentiment, **regardless** of the test query content.

to generate adversarial examples, which may not be feasible in real-world applications. Crucially, existing attacks are not specifically crafted to target techniques based on LLMs, i.e., ICL. As such, the inherent security risks of LLMs remain largely unexplored. There is an urgent need for red teaming tailored to ICL to expose the substantial risk of LLMs for further evaluating their adversarial robustness against potential real-world threats.

1.2 Adversarial Attack on ICL

This work proposes a novel adversarial attack specifically targeting ICL. We develop a gradient-based prompt search algorithm to learn adversarial suffixes in order to efficiently and effectively hijack LLMs via adversarial ICL. [**?**, **?**] are the closest works to ours where

they 'search' adversarial examples to simply manipulate model outputs. Yet, our attack method 'learns' adversarial tokens that directly hijack LLMs to generate the unwanted target that disrupts alignment with the desired output. In Figure **??**, we illustrate the major difference between the previous attack and the proposed attack: ours hijacks LLMs to output the unwanted target response (e.g.,'negative') regardless of the query content. Furthermore, instead of manipulating the prompt instructions [**?**, **?**], demos [**?**, **?**], or queries [**?**, **?**] leveraging standard adversarial examples, e.g., character-level attacks [**?**, **?**], which are detectable easily, our hijacking attack is imperceptible in that it adds only 1-2 suffixes to the demos, as shown in Figure **??**. Specifically, these suffixes are semantically incongruous but not easily identified as typos or gibberish compared to the existing ICL attack [**?**]. Finally, the backdoor attack during ICL [**?**] requires a trigger, which is impractical in real-world scenarios, whereas our attack hijacks the LLM to generate the unwanted target without triggering the queries.

Our extensive experiments validate the efficacy of our hijacking attacks. First, the attacks reliably induce LLMs to generate the targeted and misaligned output from the desired ones. Second, the adversarial suffixes learned via gradient optimization are transferable, remaining effective on different demo sets. Third, the suffix transferability holds even across different datasets for the same downstream task. Finally, our analysis shows that the adversarial suffixes distract LLMs' attention away from the task-relevant concepts. Our adversarial ICL attack poses a considerable threat to practical LLM applications due to its robust transferability and imperceptibility.

1.3 Defense against Adversarial Attack on ICL

As this work represents one of the first efficient adversarial demonstration attacks during ICL, strategies for defending against such attacks have yet to be thoroughly investigated. Recently, [**?**] introduced a method for defending against back-door attacks at test time, leveraging few-shot demonstrations to correct the inference behavior of poisoned LLMs. Similarly, [**?**] explored the power of in-context demos in manipulating the alignment ability of LLMs and proposed in-context attack and in-context defense methods for jailbreaking and guarding the aligned LLMs. Consequently, we explore the potential of using in-context demos exclusively to rectify the behavior of LLMs subjected to our hijacking attacks. Our defense strategy employs additional clean in-context demos at test time to safeguard LLMs from being hijacked by adversarial in-context demos. The experimental results demonstrate the efficacy of our proposed defense method against various adversarial demonstration attacks.

1.4 Contribution

This work makes the following original contributions: (1) We propose a novel stealthy adversarial attack targeting in-context demos to hijack LLMs to generate unwanted target output during ICL. (2) We design a novel gradient-based prompt search algorithm to learn and append adversarial suffixes to the in-context demos. (3) Our extensive experiments demonstrate the effectiveness and transferability of the proposed adversarial ICL attack across various demo sets, LLMs, and tasks. (4) The proposed test time defense strategy effectively protects LLMs from being compromised by adversarial demonstration attacks.

CHAPTER 2 RELATED WORK

2.1 In-Context Learning

LLMs have shown impressive performance on numerous NLP tasks [**?**, **?**, **?**]. Although fine-tuning has been a common method for adapting models to new tasks, it is often less feasible to fine-tune extremely large models with over 10 billion parameters. As an alternative, recent work has proposed ICL, where the model adapts to new tasks solely via inference conditioned on the provided in-context demos, without any gradient updates [**?**]. By learning from the prompt context, ICL allows leveraging massive LLMs' knowledge without the costly fine-tuning process, showcasing an exemplar of the LLMs' emergent abilities [**?**, **?**].

Intensive research has been dedicated to ICL. Initial works attempt to find better ways to select labeled examples for the demos [**?**, **?**]. For instance, [**?**] presents a simple yet effective retrieval-based method that selects the most semantically similar examples as demos, leading to improved accuracy and higher stability. Follow-up works have been done to understand why ICL works [**?**, **?**, **?**, **?**, **?**]. [**?**] provides theoretical analysis that ICL can be formalized as Bayesian inference that uses the demos to recover latent concepts. Another line of research reveals the brittleness and instability of ICL approaches: small changes to the demo examples, labels, or order can significantly impact performance [**?**, **?**, **?**, **?**].

2.2 Adversarial Attacks on LLMs

Early adversarial attacks on LLMs apply simple character or token operations to trigger the LLMs to generate incorrect predictions, such as TextAttack [**?**] and BERT-Attack [**?**]. Since these attacks usually generate misspelled and/or gibberish prompts that can be

detected using spell checker and perplexity-based filters, they are easy to block in real-world applications. Some other attacks struggled with optimizing over discrete text, leading to the manual or semi-automated discovery of vulnerabilities through trial-and-error [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**]. For example, jailbreaking prompts are intentionally designed to bypass an LLM's built-in safeguard, eliciting it to generate harmful content that violates the usage policy set by the LLM vendor [**?**, **?**, **?**, **?**, **?**, **?**, **?**]. These red teaming efforts craft malicious prompts in order to understand LLM's attack surface [**?**]. However, the discrete nature of text has significantly impeded learning more effective adversarial attacks against LLMs.

Recent work has developed gradient-based optimizers for efficient text modality attacks. For example, [**?**] presented a gradient-based discrete optimizer that is suitable for attacking the text pipeline of CLIP, efficiently bypassing the safeguards in the commercial platform. [**?**], building on [**?**], described an optimizer that combines gradient guidance with random search to craft adversarial strings that induce LLMs to respond to the questions that would otherwise be banned. Subsequently, [**?**] utilized the optimization algorithm from [**?**] to identify the minimal prompt that elicits the target output, which may reveal private or copyrighted content, in the zero-shot setting of ICL. This method could also circumvent certain safeguards, such as in-context unlearning [**?**], to provoke harmful responses. More recently, [**?**] proposed poisoning demonstration examples and prompts to make LLMs behave in alignment with pre-defined intentions.

Our hijacking attack algorithm falls into this stream of work, yet we target few-shot ICL instead of zero-shot queries. We use gradient-based prompt search to automatically learn effective adversarial suffixes rather than manually engineered prompts. Importantly, we show that LLMs can be hijacked to output the targeted unwanted output by appending opti-

mized adversarial tokens to the ICL demos, which reveals a new lens of LLM vulnerabilities that may have been missed by prior approaches.

2.3 Defense Against Attacks on LLMs

The existing literature on the robustness of LLMs includes various strategies for defense [**?**, **?**, **?**, **?**, **?**]. However, most of these defenses, such as those involving adversarial training [**?**, **?**, **?**, **?**] or data augmentation [**?**, **?**], need to re-train or fine-tune the models, which is computationally infeasible for LLM users. Moreover, the restriction of many closed-source LLMs to only permit query access for candidate defenses introduces new challenges.

Recent studies focus on developing defenses against attacks on LLMs that utilize adversarial prompting. [**?**] and [**?**] have suggested the use of perplexity filters to detect adversarial prompts. While the filters are effective at catching the attack strings that contain gibberish words or character-level adversarial tokens with high perplexity scores, they fall short in detecting more subtle adversarial prompts, like the ones used in our adversarial demonstration attacks with as low perplexity as clean samples shown in Figure **??**. Recently, [**?**] introduced a method to mitigate backdoor attacks at test time by identifying the task and retrieving relevant defensive demonstrations. These demonstrations are combined with user queries to counteract the adverse effects of triggers present in backdoor attacks. This defense strategy eliminates the need for modifications or tuning of LLMs. Its objective is to re-calibrate and correct the behavior of LLMs during test-time evaluations. Similarly, [**?**] investigated the role of in-context demonstrations in enhancing the robustness of LLMs and highlighted their effectiveness in defending against jailbreaking attacks. The authors developed an in-context defense strategy that constructs a safe context to caution the model against generating any harmful content.

So far, defense mechanisms against adversarial demonstration attacks have not been extensively explored. Our approach introduces a test-time defense strategy that uses additional clean in-context demos to safeguard LLMs from adversarial in-context manipulations. In line with prior works [**?**, **?**, **?**], this defense strategy avoids the necessity for retraining or fine-tuning LLMs. Instead, it focuses on re-calibrating and correcting the behavior of LLMs during evaluations at test time.

CHAPTER 3 PRELIMINARIES

3.1 ICL Formulation

Formally, ICL is characterized as a problem involving the conditional generation of text [**?**], where an LLM $\mathcal{M}$ is employed to generate response y_Q given an optimal task instruction I, a demo set C, and an input query x_Q. I specifies the downstream task that $\mathcal{M}$ should perform, e.g., "Choose sentiment from positive or negative" used in our sentiment generation task. C consists of N (e.g., 8) concatenated data-label pairs following a specific template S, formally: $C = [S(x_1, y_1);\ \cdots\ ;\ S(x_N, y_N)]$, ';' here denotes the concatenation operator. Thus, given the input prompt as $p = [I;\ C;\ S(x_Q, \_)]$, $\mathcal{M}$ generates the response as $\hat{y}_Q = \mathcal{M}(p)$. $S(x_Q, \_)$ here means using the same template as the demos but with the label empty.

3.2 Adversarial Attack on LLMs

In typical text-based adversarial attacks, the attackers manipulate the input x with the goal of misleading the model to produce inaccurate output or bypass safety guardrails [**?**, **?**]. Specifically, given the input-output pair (x, y), the attackers aim to learn the adversarial perturbation δ adding to x by maximizing the model's objective function but without misleading humans by bounding the perturbation within the "perceptual" region Δ. The objective function of the attacking process thus can be formulated as:

$$\max_{\delta \in \Delta} \mathcal{L}(\mathcal{M}(x_Q + \delta), y_Q), \tag{3.1}$$

$\mathcal{L}$ here denotes the task-specific loss function, for instance, cross-entropy loss for classification tasks.

CHAPTER 4 THE THREAT MODEL

4.1 ICL Hijacking Attack

ICL consists of an instruction I, a demo set C, and an input query x_Q, providing more potential attack vectors compared to the conventional text-based adversarial attacks. This work focuses on manipulating C without changing I and x_Q. Recently, [**?**] applied character-level perturbation techniques (e.g., TextAttack [**?**] and TextBugger [**?**]), such as character insertion, character deletion, neighboring character swap, and character substitution, to reverse the output.

Our hijacking attack learns the adversarial suffix tokens to the in-context demos to manipulate LLMs' output via a new greedy gradient-based prompt injection algorithm. Given a clean demo set:

$$C = [S(x_1, y_1);\ \cdots\ ;\ S(x_N, y_N)], \tag{4.1}$$

our hijacking attack automatically produces an adversarial suffix for each demo in c, formally:

$$C' = [S(x_1 + \delta_1, y_1);\ \cdots\ ;\ S(x_N + \delta_N, y_N)], \tag{4.2}$$

where C' denotes the perturbed demo set. To make it clear, the adversarial suffixes appended to each demo as perturbations are different, respectively. In this case, the attack or perturbation budget refers to the number of tokens in each adversarial suffix.

As a result, our hijacking attack induces $\mathcal{M}$ to generate an unwanted target output y_T via appending adversarial suffix tokens on the in-context demos as $y_T = \mathcal{M}(p')$. In other words, $\mathcal{M}$ generates the same or different responses for the clean and perturbed prompts

depending on the True or False of $\mathcal{M}(p) = y_T$:

$$\begin{cases} \mathcal{M}(p) = \mathcal{M}(p'), & \text{True}, \\ \mathcal{M}(p) \neq \mathcal{M}(p'), & \text{False}, \end{cases}$$

where $p = [I;\ C;\ S(x_Q, \_)]$ and $p' = [I;\ C';\ S(x_Q, \_)]$, respectively.

4.2 Hijacking Attack Objective

We express the goal of the hijacking attack as a formal objective function. Let us consider the LLM $\mathcal{M}$ as a function that maps a sequence of tokens $x_{1:n}$, with $x \in \{1, \cdots, V\}$ where V denote the vocabulary size, namely, the number of tokens, to a probability distribution over the next token x_{n+1}. Specifically, $\mathcal{P}(x_{n+1}|x_{1:n})$ denotes the probability that x_{n+1} is the next token given the previous tokens $x_{1:n}$. In more detail, we formulate $x_{1:n}$ as $[I;\ C;\ S(x_Q, \_)]$ and x_{n+1} as $\hat{y}_Q$ in the context of an ICL task, respectively.

Using the notations defined earlier, the hijacking attack objective we want to optimize is simply the negative log probability of the target token x_{n+1}. The generated target output y_T is different from the ground truth label y_Q for the training query (x_Q, y_Q). Formally:

$$\mathcal{L}(x_Q) = -\log \mathcal{P}(\mathcal{M}(y_T|p')), \tag{4.3}$$

where $y_T \neq y_Q$, demonstrating the attack hijacks $\mathcal{M}$ to generate the target output. p' denotes the perturbed prompt as: $p' = [I;\ C';\ S(x_Q, \_)]$. In summary, the problem of optimizing the adversarial suffix tokens can be formulated as the following optimization

objective:

$$\underset{\delta_i \in \{1,\cdots,V\}^{|N|}}{\text{minimize}} \mathcal{L}(x_Q), \tag{4.4}$$

where i denotes the indices of the demos and N is the number of demos in the perturbed demos set C', respectively.

4.3 Greedy Gradient-guided Injection

A primary challenge in optimizing Eq. **??** is optimizing over a discrete set of possible token values. While there are some methods for discrete optimization, prior work [**?**] has shown that those effective strategies often struggle to reliably attack the aligned LLMs.

Motivated by prior works [**?**, **?**, **?**], we propose a simple yet effective algorithm for LLMs hijacking attacks, called greedy gradient-guided injection (GGI) algorithm (Algorithm **??**). The key idea comes from greedy coordinate descent: if we could evaluate all possible suffix token injections, we could substitute the tokens that maximize the adversarial loss reduction. Since exhaustively evaluating all tokens is infeasible due to the large candidate vocabulary size, we instead leverage gradients with respect to the suffix indicators to find promising candidate tokens for each position. We then evaluate all of these candidate injections with explicit forward passes to find the one that decreases the loss the most. This allows an efficient approximation of the true greedy selection. We can optimize the discrete adversarial suffixes by iteratively injecting the best tokens.

We compute the linearized approximation of replacing the demo x_i in C by evaluating the gradient $\nabla_{\mathbf{e}_{x_i^j}} \mathcal{L}(x_Q) \in \mathbb{R}^{|V|}$, where $\mathbf{e}_{x_i^j}$ denotes the vector representing the current value of the j-th adversarial suffix token. Note that because LLMs typically form embeddings for each token, they can be written as functions of $\mathbf{e}_{x_i^j}$, and thus we can immediately take the

Algorithm 1: Greedy Gradient-guided Injection (GGI)

Input : Model: $\mathcal{M}$, Iterations: T, Batch Size: b, Instruction: I, Demos: C, Query: (x_Q, y_Q) Target: y_T

Initialization: $p'_0 = [I;\ [S(x_1 + \delta_1, y_1);\ \cdots;\ S(x_N + \delta_N, y_N)];\ S(x_Q, y_T)]$

repeat

  for $i \in N$ **do**

    $[\delta_{i_1}; ...; \delta_{i_k}] = \text{Top}{-}k(-\nabla_{p'}\mathcal{L}(\mathcal{M}(\hat{y}|p'_{t-1}), y_T))$ */* Compute top-k substitutions */*

  $K = \{[\delta_{i_1};\ ...;\ \delta_{i_k}] \mid i = 1,\ ..., N\}$

  $B = \{(\delta_{i1}, \ldots, \delta_{ib}) \mid (\delta_{i1}, \ldots, \delta_{ik}) \in K\}$ */* Make a subset of substitution */*

  for $i \in N$ **do**

    $\delta_i^\star = \delta_{ij}$, where $j = \text{argmin}_{\delta_{ib}}\mathcal{L}(\mathcal{M}(\hat{y}|p'_{t-1}), y_T)$ */* Compute best replacement */*

  $\Delta = [\delta_1^\star;\ ...;\ \delta_N^\star]$

  $p'_t = [I;\ [S(x_1 + \delta_1^\star, y_1);\ \cdots;\ S(x_N + \delta_N^\star, y_N)];\ S(x_Q, y_T)]$ */* Update prompt */*

until T *times*;

Output : Optimized prompt suffixes $[\delta_1^\star, \cdots, \delta_N^\star]$

gradient with respect to this quantity [**?**, **?**].

The key aspects of our GGI algorithm (Algorithm **??**) are: firstly, it uses gradients of the selected token candidates to calculate the top candidates; secondly, it evaluates the top candidates explicitly to identify the most suitable one; and lastly, it iteratively injects the best token at each position to optimize the suffixes. This approximates an extensive greedy search in a computationally efficient manner.

CHAPTER 5 THE DEFENSE METHOD

Having developed the ICL hijacking attack by incorporating adversarial tokens into the in-context demos, we now present a straightforward yet potent defense strategy to counter this attack. Initially, we assume that defenders treat LLMs as black-box, lacking any insight into their training processes or underlying parameters. The defenders apply defense on the input prompt p directly during test-time evaluation. Their goal is to rectify the behavior of LLMs and induce LLMs to generate desired responses to user queries.

Given an input prompt p' that includes adversarial tokens within the demos C', we assume that LLMs, when presented with demos containing clean data for the same tasks, can understand the genuine intent of the user's query through ICL, rather than being misled by the adversarial demos. In this context, 'clean data' refers to data without any adversarial tokens and is randomly selected from the training set. More precisely, the defenders modify the input prompt p' into $\tilde{p}$ by appending or inserting more clean demos into the demo set C', as follows: $\tilde{p} = [I; C'; \tilde{C}; S(x_Q, \_)]$. $\tilde{C} = [S(\tilde{x}_1, \tilde{y}_1);\ \cdots\ ;\ S(\tilde{x}_N, \tilde{y}_N)]$ here denotes the clean demos selected from the training set. Through this approach, the defender guarantees that the in-context demos align with the user's query and possess resilience against adversarial attacks. In our experiments, we maintained an equal number of demos in C' and $\tilde{C}$ and observed that this method resulted in effective defense across a range of datasets and tasks.

CHAPTER 6 EXPERIMENT SETUP

6.1 Datasets

We evaluate the performance of our LLM hijacking algorithm and other baseline algorithms on three classification datasets covering sentiment analysis and topic generation tasks. SST-2 [**?**] and Rotten Tomatoes (RT) [**?**] are both binary sentiment analysis datasets of movie reviews. AG's News [**?**] is a multi-class news topic classification dataset. These datasets allow us to evaluate the hijacking attacks on a diverse set of text classification benchmarks across both binary and multi-class settings.

We show the dataset statistics in Table **??**. Specifically for the SST-2 and RT sentiment analysis tasks, we employ only 2 training queries to train adversarial suffixes using our GGI method. We use 4 training queries for the more complex multi-class topic generation tasks, i.e., AG's News. We randomly select 1,000 samples as user queries for testing.

Table 1: Statistics of the training queries used in Algorithm **??** and test queries for the three datasets.

Datasets	Training Queries	Test Queries
SST-2	2	1,000
RT	2	1,000
AG's News	4	1,000

6.2 Large Language Models

The experiments are conducted using three different LLMs, GPT2-XL [**?**], LLaMA-7b/13b [**?**], OPT-2.7b/6.7b [**?**], and Vicuna-7b [**?**] allowing us to evaluate attack effectiveness on both established and SOTA LLMs. The choice of LLMs covers a diverse set of architectures and model sizes, enabling a comprehensive evaluation of the vulnerability of LLMs via adversarial ICL.

Dataset	Structure	Template	Example
SST-2/RT	Instruction	[instruction]	Analyze the sentiment of the last review and respond with either positive or negative. Here are several examples.
	Demos	Review: [sentence] Sentiment: label … Review: [sentence] Sentiment: [label]	Review: "a retread story , bad writing , and the same old silliness " Sentiment: negative … Review: is definitely a director to watch Sentiment: positive
	Queries	Review: [sentence] Sentiment:	Review: waste their time on it Sentiment:
AG-News	Instruction	[instruction]	Classify the topic of the last article. Here are several examples.
	Demos	Article: [article] Topic: [label] … Article: [article] Topic: [label]	Article: Skier Tests Positive Olympic silver medalist Hans Knauss tests positive for the steroid nandrolone after a World Cup race last month. Topic: sports … Article: "Apple Unwraps New IMac G5s PARIS -- Apple Computer will begin shipping its new IMac G5 desktop computer worldwide in mid-September, the company #39;s top marketing executive says.", Topic: technology
	Queries	Article: [article] Topic:	Article: Microsoft adds to Visual Studio tools line 2005 Standard Edition targets developers working in small organizations. Topic:

Figure 2: Template designs for all the datasets used in our experiments. We also provide examples for these datasets in Figure **??** and Figure **??** to ensure a better understanding.

6.3 ICL Settings

For ICL, we follow the setting in [**?**] and use their template to incorporate the demos for prediction. Figure **??** illustrates the prompt template employed in ICL for various tasks. For the SST2/RT dataset, the template is structured to include an instruction, a demo set composed of reviews and sentiment labels, and the user query. Similarly, the AG's News dataset template comprises the instruction, the demo set with articles and topic labels, and the user query. Additionally, examples are provided in Figure **??** and Figure **??** to enhance understanding.

We evaluate the 2-shot, 4-shot, and 8-shot settings for the number of demos. Specifically,

for each test example, we randomly select the demos from the training set and repeat this process 5 times, reporting the average accuracy over the repetitions.

6.4 Evaluation Metrics

Several different metrics evaluate the performance of ICL and hijacking attacks. Clean accuracy evaluates the accuracy of ICL on downstream tasks using clean demos. Attack accuracy evaluates the accuracy of ICL given the perturbed demons. Defense accuracy demonstrates the accuracy of ICL with the defense method against the hijacking attack. We further evaluate the effectiveness of hijacking attacks using attack success rate (ASR). Given a test sample (x, y) from a test set D, the clean and perturbed prompts are denoted as $p = [I; C; x]$ and $p' = [I; C'; x]$, respectively. ASR is calculated as:

$$\text{ASR} = \sum_{(x,y)\in D} \frac{\mathbb{1}(\mathcal{M}(p') = y_T)}{\mathbb{1}(\mathcal{M}(p) = y)}, \tag{6.1}$$

where $\mathbb{1}$ denotes the indicator function.

6.5 Baseline Attacks

6.5.1 Greedy Search

We consider a heuristics-based perturbation strategy, which conducts a greedy search over the vocabulary to select tokens, maximizing the reduction in the adversarial loss from Eq. **??**. Specifically, it iteratively picks the token that decreases the loss the most at each step.

6.5.2 Square Attack

The square attack [**?**] is an iterative algorithm for optimizing high-dimensional black-box functions using only function evaluations. To find an input $x + \delta$ in the demo set C that minimizes the loss in Eq. **??**, the square attack has three steps: Step 1: Select a subset of inputs to update; Step 2: Sample candidate values to substitute for those inputs; Step 3: Update $x + \delta$ with the candidate values that achieve the lowest loss. The square attack can optimize the hijacking attack objective function without requiring gradient information by iteratively selecting and updating a subset of inputs.

6.5.3 Text Attack

We also utilize TextAttack (TA) [**?**], adopting a similar approach to the attack described by [**?**], which serves as the most closely related baseline for our hijacking attack. Different from our word-level attack, the use of TA at the character level includes minor modifications to some words in the in-context demos and simply flips the labels of user queries, as depicted in Figure **??**. In our experiments, we employ a transformation where characters are swapped with those on adjacent QWERTY keyboard keys, mimicking errors typical of fast typing, as done in TextAttack [**?**]. Specifically, we use the adversarial examples for the same demos in our hijacking attack during the application of TA.

CHAPTER 7 RESULT AND DISCUSSION

7.1 ICL Performance

The rows identified as 'Clean' in Table **??** and Table **??** show the ICL performance on the respective tasks when using clean in-context demos. In particular, Table **??** presents the accuracies for the generation of positive (P) and negative (N) sentiments in the SST-2 and RT datasets. All the tested LLMs perform well, achieving an average accuracy of 83.6% on SST-2 and 86.7% on RT across various in-context few-shot settings. Table **??** indicates that LLMs with ICL also perform well in the context of multi-class generation on AG's News dataset. The average accuracies stand at 69.1% for 4-shot settings and 72.3% for 8-shot settings across various LLMs. Additionally, LLMs with ICL exhibit improved performance with an increased number of in-context demos, particularly achieving best results with 8-shot settings.

7.2 ICL Performance with Hijacking Attack

While LLMs utilizing ICL show strong performance with clean in-context demos, Tables **??** and **??** reveal that their effectiveness is significantly undermined by hijacking attacks. These attacks manipulate the models to produce target outputs by appending adversarial suffixes to the in-context demos. The baseline attacks successfully induce LLMs to generate the targeted positive sentiment through a few shots of adversarially perturbed demos, as shown in Table **??**. As a result, the positive test samples achieve a predominantly higher accuracy than the negative ones. Furthermore, our GGI attack more effectively hijacks LLMs to generate the target output, i.e., exclusively positive sentiment token, regardless of the test query content. The positive test samples achieve almost 100% accuracy. On

Table 2: The performance on sentiment analysis task with and without attacks on ICL. The row identified as 'Clean' in gray color represents the accuracy with clean in-context demos. Other rows illustrate the accuracies with adversarial in-context demos. The details of the baselines in green color are present in Section **??**. Specifically, we employ TextAttack (TA) [**?**] following the attack in [**?**] as the most closely related baseline for our attack (GGI). The accuracies of positive (P) and negative (N) sentiments are reported separately to highlight the effectiveness of our hijacking attack.

Model	Method	SST-2						RT					
		2-shots		4-shots		8-shots		2-shots		4-shots		8-shots	
		P	N	P	N	P	N	P	N	P	N	P	N
GPT2-XL	Clean	94.7	52.2	88.6	49.4	91.6	69.0	93.3	54.7	88.6	76.9	90.2	80.5
	Square	99.4	2.0	99.8	4.2	99.4	11.0	99.8	1.5	100	4.1	99.3	7.5
	Greedy	100	10.8	100	6.2	100	0.2	100	5.3	100	2.8	100	0.0
	TA	95.0	2.2	99.8	17.8	99.6	21.6	95.9	8.1	96.3	41.3	96.4	47.3
	GGI	**100**	**1.2**	**100**	**0.0**	**100**	**0.0**	**100**	**2.8**	**100**	**0.0**	**100**	**0.0**
OPT-2.7b	Clean	98.5	38.6	85.6	62.8	58.4	76.4	98.1	36.6	81.2	68.4	57.8	89.6
	Square	100	0.0	100	0.0	100	1.8	100	1.3	100	0.0	99.6	7.5
	Greedy	100	0.0	100	0.0	100	0.0	100	0.4	100	0.2	100	0.0
	TA	99.6	13.8	99.8	26.8	99.0	7.2	97.6	52.9	97.2	59.7	99.4	6.8
	GGI	**100**	**0.0**	**100**	**0.0**	**100**	**0.0**	**100**	**0.0**	**100**	**0.0**	**100**	**0.0**
OPT-6.7b	Clean	69.4	87.8	70.2	93.8	77.8	93.0	84.4	91.4	84.4	93.1	88.6	92.8
	Square	99.2	31.4	93.8	72.2	99.6	29.0	98.1	42.2	97.0	68.7	99.4	33.2
	Greedy	100	25.0	97.8	39.0	100	2.0	99.4	31.7	99.8	4.7	100	0.8
	TA	94.8	80.8	54.8	98.6	91.6	89.4	92.5	86.1	77.6	96.4	94.0	86.3
	GGI	**100**	**0.0**	**98.4**	**2.0**	**100**	**0.2**	**100**	**2.6**	**99.8**	**0.0**	**100**	**0.2**
Vicuna-7b	Clean	91.4	81.2	88.2	81.4	94.6	82.6	84.8	78.4	85.9	80.5	90.4	85.4
	Square	89.2	84.4	86.6	85.8	94.0	83.8	85.9	85.4	84.6	88.6	91.6	88.4
	Greedy	93.0	83.4	88.4	87.0	94.6	80.0	91.2	82.8	86.9	88.7	91.9	85.9
	TA	87.0	85.2	76.2	88.2	94.2	80.6	83.3	84.2	79.6	88.6	92.1	84.4
	GGI	**90.6**	**42.2**	**96.4**	**23.2**	**100**	**0.8**	**87.6**	**36.4**	**95.1**	**35.7**	**100**	**0.2**
LLaMA-7b	Clean	81.4	86.3	74.4	91.9	82.7	92.4	86.0	83.6	81.9	91.6	89.3	97.8
	Square	86.8	80.0	96.8	58.6	98.0	56.4	86.9	57.4	97.4	50.1	97.8	57.4
	Greedy	95.0	47.6	100	0.0	100	0.0	88.9	2.8	99.8	0.0	100	0.0
	TA	87.2	77.8	93.8	69.0	99.8	8.8	83.1	57.4	94.2	68.9	99.6	3.80
	GGI	**100**	**0.4**	**100**	**0.0**	**100**	**0.0**	**96.8**	**0.0**	**100**	**0.0**	**100**	**0.0**
LLaMA-13b	Clean	97.8	76.4	95.6	88.0	95.8	90.0	94.2	84.8	92.7	92.1	91.4	91.9
	Square	98.4	72.8	98.2	78.4	97.8	85.4	93.6	87.4	94.4	84.1	94.2	87.6
	Greedy	98.0	41.4	100	3.0	100	0.0	55.9	11.3	92.9	0.0	100	0.4
	TA	98.2	72.2	92.8	92.8	97.5	87.6	94.8	81.8	88.0	94.0	92.5	89.3
	GGI	**99.2**	**37.8**	**100**	**7.2**	**100**	**0.0**	**99.1**	**3.8**	**86.1**	**3.6**	**100**	**0.0**

the contrary, the negative ones get nearly 0% accuracy on most settings, as shown in **??**. For the more complex multi-class AG's News topic generation task, the effectiveness of those baseline attacks decreases significantly. Especially the TA method is not successful in hijacking LLMs to generate the target token, i.e., 'tech'. Only our GGI attack successfully hijacks the LLMs to generate "tech" by appending the adversarial suffixes to the in-context demos, as shown in Table **??**.

In addition to the attack accuracy performance provided in Table **??** and **??**, we present ASRs for various attacks across the three datasets. As outlined in Table **??**, our GGI attack achieves the highest ASRs, substantiating its highest effectiveness in hijacking the LLM to

Table 3: The performance of AG's News topic generation task with and without attacks on ICL. The clean and attack accuracies are reported separately for the four topics. These results highlight the effectiveness of our hijacking attacks to induce LLMs to generate the target token, i.e., "tech", regardless of the query content.

Model	Method	4-shots				8-shots			
		word	sports	business	**tech**	word	sports	business	**tech**
GPT2-XL	Clean	48.5	87.0	64.9	71.9	48.2	50.6	71.0	83.6
	Square	2.0	66.0	26.8	96.0	19.6	65.6	28.0	97.2
	Greedy	12.8	60.4	29.2	96.4	8.0	21.2	10.0	98.8
	TA	54.8	84.0	73.2	82.4	82.0	82.4	91.2	57.6
	GGI	**0.0**	**2.0**	**0.4**	**100**	**0.0**	**0.0**	**0.0**	**100**
LLaMA-7b	Clean	68.2	96.8	66.6	49.0	88.6	97.4	78.2	61.0
	Square	78.4	98.0	76.0	36.8	94.4	98.0	60.0	57.6
	Greedy	69.6	98.8	75.2	51.6	89.6	100	68.4	73.6
	TA	42.4	94.8	67.6	32.4	95.2	96.0	39.2	24.8
	GGI	**0.0**	**20.0**	**0.00**	**98.0**	**29.6**	**56.0**	**0.0**	**100**

Table 4: ASR among different datasets, models, and attack methods. Best scores are in bold.

Model	Method	SST-2			RT			AG's News	
		2-shots	4-shots	8-shots	2-shots	4-shots	8-shots	4-shots	8-shots
GPT2-XL	Square	98.0	97.8	94.2	**98.7**	97.9	95.9	64.9	65.2
	Greedy	94.6	96.9	99.9	97.4	98.6	100	68.3	87.3
	TA	89.6	91.0	89.0	85.9	77.5	74.6	15.1	15.9
	GGI	**99.4**	**100**	**100**	98.6	**100**	**100**	**99.1**	**100**
LLaMA-7b	Square	48.1	65.9	70.6	48.4	69.9	69.7	10.3	15.9
	Greedy	64.2	100	100	64.3	99.8	100	14.3	22.1
	TA	48.2	59.5	95.4	45.8	58.0	97.8	9.3	6.8
	GGI	**97.7**	**100**	**100**	**90.7**	**99.9**	**100**	**82.8**	**77.9**

generate the targeted output. In sentiment analysis tasks like SST-2 and RT, all attacks exhibit high ASRs. While for the more complex multi-class topic generation task, such as AG's News, only our GGI attack achieves high ASRs. This further emphasizes the potential effectiveness of our hijacking attack on more complex generative tasks, such as question answering tasks.

7.3 Defense Method Performance

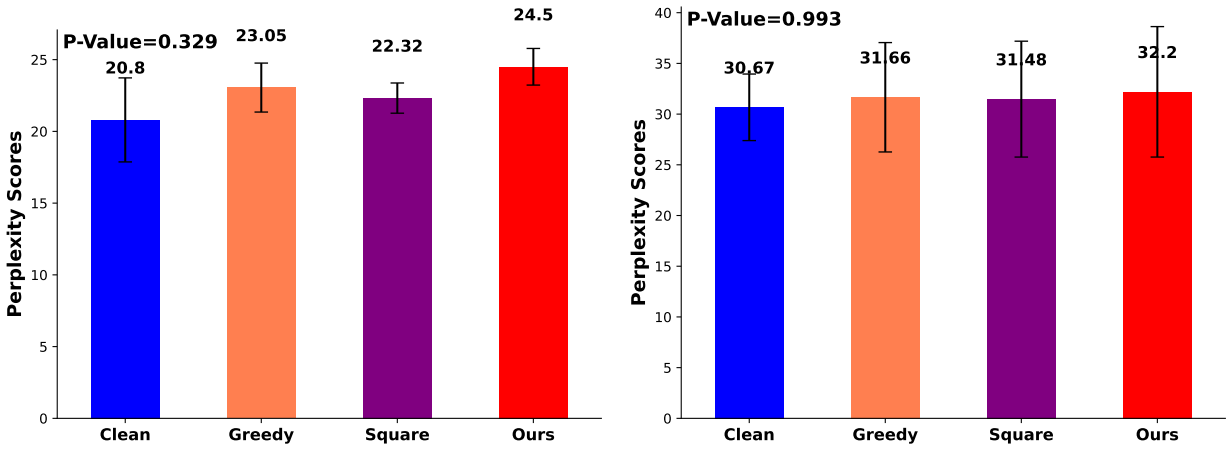
Table **??** presents ASRs of our hijacking attack when countered with the proposed defense mechanism that uses additional clean demos. The defense method is tested in two different settings. In the Preceding setting, clean demos are placed before the adversarial demos in the sequence $\tilde{p} = [I; \tilde{C}; C'; S(x_Q, \_)]$. Conversely, in the Proceeding setting, clean demos are added after the adversarial demos, forming the sequence $\tilde{p} = [I; C'; \tilde{C}; S(x_Q, \_)]$.

Table 5: The performance of the defenses using ASRs across various LLMs and datasets. Adv denotes our hijacking attack using the adversarial demos. Adv+Clean represents the proposed defense method, leveraging extra clean demos with adversarial demos. The numbers within the parenthesis indicate the reduction in the ASRs after defense.

Model	**SST-2**			**RT**			**AG's News**		
	Adv	Adv+Clean		Adv	Adv+Clean		Adv	Adv+Clean	
		Preceding	Proceeding		Preceding	Proceeding		Preceding	Proceeding
GPT2-XL	100	100 (-0)	99.6 (-0.4)	100	100 (-0)	97.4 (-2.6)	99.1	75.5 (-23.6)	80.5 (-18.6)
OPT-6.7b	98.2	44.9 (-53.3)	52.5 (-45.7)	99.9	50.2 (-49.7)	57.8 (-42.1)	65.6	23.5 (-42.1)	22.5 (-43.1)
LLaMA-7b	100	49.1 (-50.9)	98.3 (-1.7)	100	53.1 (-46.9)	99.8 (-0.2)	82.8	42.2 (-40.6)	88.2 (+5.4)

The results show a significant decrease in ASRs of our hijacking attack, affirming the effectiveness of the defense method. Notably, the Preceding setting results in considerably lower ASRs compared to the Proceeding setting. This relates to the mechanism through which our hijacking attack induces LLMs to generate target outputs. As depicted in Figure **??**, the adversarial suffixes divert the LLMs' attention away from the original query. Furthermore, Figure **??** illustrates that the LLM primarily focuses on the initial segments of the demos, which are indicated by a darker green color. Therefore, in the Preceding method, the model shifts its attention to these first few demos, which contain additional clean samples before the adversarial demos. These clean samples effectively re-calibrate and rectify the model's behavior, resulting in a larger reduction in ASRs, as shown in Table **??**. In contrast, the first few demos remain adversarial in the Proceeding method, rendering it ineffective in defending against the adversarial demonstration attack.

Furthermore, the results indicate that our proposed defense methods are ineffective on small-sized LLMs, such as the GPT2-XL used in our experiments. We hypothesize that this is due to their limited emergent abilities. In other words, employing additional demos during ICL cannot correct the behavior of small-sized LLMs under hijacking attacks.



(a) SST-2 (b) RT

Figure 3: Average perplexity scores reported for LLaMA2-7b on 100 random samples under eight-shots setting from SST-2 (a) and RT (b) derived from three separate runs under various attacks.

7.4 Stealthiness of GGI

The perplexity scores shown in Figure **??** for both the baseline and our attacks exhibit minor (non-significant, P-value of 0.329 (0.993) $\gg$ 0.05 cutoff is reported from one-way ANOVA test) increases compared to the score of the clean samples, highlighting the stealth of our proposed word-level adversarial attacks. Specifically, the adversarial triggers learned from our GGI algorithm are imperceptible and maintain the semantic integrity and coherence of the original content, as shown in the examples of Figures **??** and **??**.

7.5 Impact of Number of In-context Demos

We extend our investigation to explore the impact of in-context demos on adversarial ICL attacks. We observe a substantial impact on the attack performance in ICL based on the number of demos employed. As indicated in Tables **??** and **??**, an increase in the number of in-context demos correlates with a higher susceptibility of the attack to hijack LLMs, resulting in the generation of target outputs with greater ease. Specifically, in the

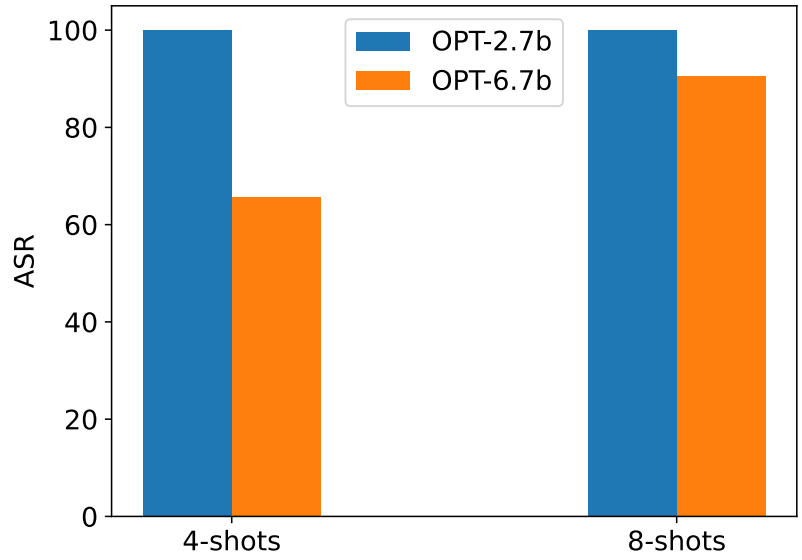


Figure 4: Impact of LLM size on adversarial robustness. ASRs on the AG's News topic generation task using different sizes of OPT models, i.e., OPT-2.7b and OPT-6.7b, with two different few-shot settings.

8-shot setting, LLMs consistently exhibit significantly lower accuracies in negative sentiment generation, demonstrating a higher rate of successful attacks compared to the 2-shot and 4-shot settings. Moreover, the attacks demonstrate higher ASRs as the number of in-context demos used in ICL increases, as shown in Table **??**.

7.6 Impact of Sizes of LLMs

Results in Table **??** reveal that the ASRs on GPT2-XL are significantly higher than those on LLaMA-7b, suggesting that hijacking the larger LLM is more challenging. Here, we continue examining how the size of LLMs influences the performance of hijacking attacks. Table **??** illustrates the performance of sentiment analysis tasks with and without attacks on ICL using different sizes of OPT, i.e., OPT-2.7b and OPT-6.7b. These results further highlight that the smaller LLM, i.e., OPT-2.7b, is much easier to be attacked and induced to generate unwanted target outputs, such as 'positive', in the sentiment analysis tasks. Figure

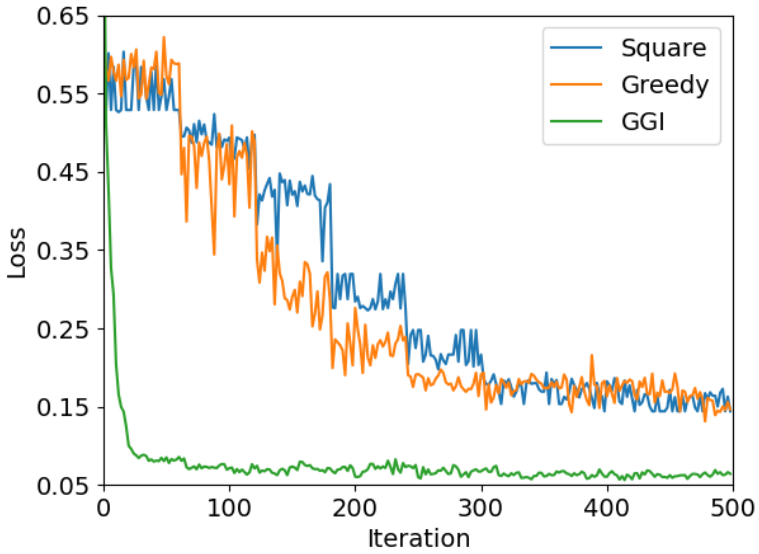


Figure 5: An illustration of the learning objective values during iterations among different attacks on SST2 using GPT2-XL with 8-shots.

?? illustrates our proposed hijacking attack performance using ASR on two OPT models of varying sizes in AG's News topic generation task. It clearly shows that attacking the smaller OPT2-2.7b model achieves a much higher ASR in both settings, confirming our finding and others [**?**] that larger models are more resistant to adversarial attacks.

7.7 Comparison of Hijacking Attacks

In contrast to baseline hijacking attacks, i.e., Square and Greedy, our GGI exhibits superior performance in generating targeted outputs, as evidenced by the results in Table **??** and **??**, along with the highest ASRs highlighted in Table **??**. This underscores the effectiveness of GGI as a more potent method of attack.

To further illustrate the efficiency of our GGI, we present the objective function values of Eq. **??** in Figure **??** for various attack methods. Since our GGI attack enjoys the advantages of both greedy and gradient-based search strategies as depicted in Algorithm **??**, the values

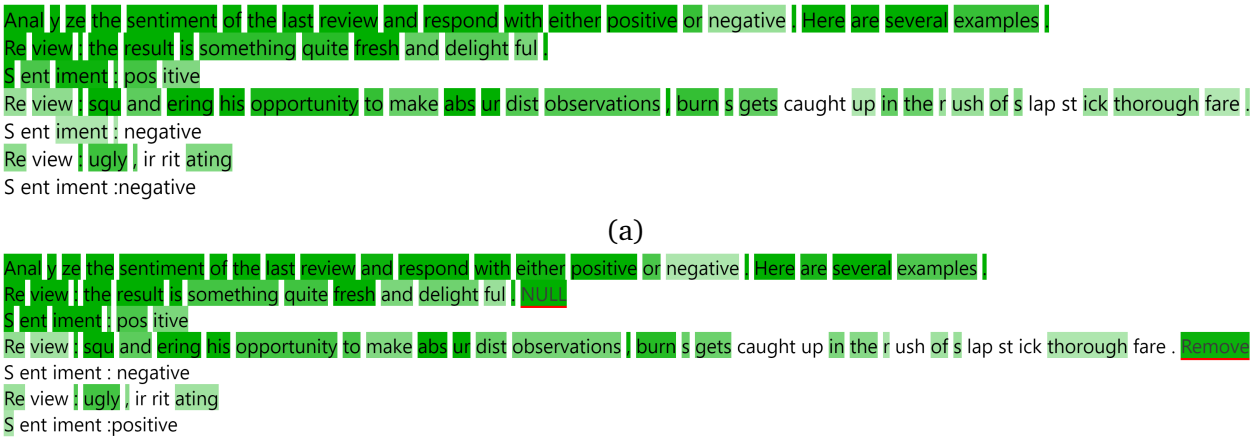
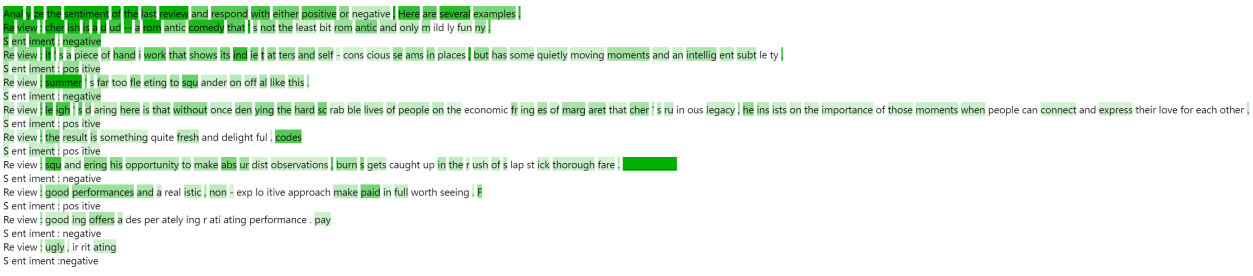


Figure 6: Attentions maps generated using (a) clean and (b) adversarial perturbed prompts. In (b), the adversarial suffix tokens, i.e., 'NULL' and 'Remove', are underlined in red. Darker green colors represent larger attention weights. The prompts are tokenized to mimic the actual inputs to the LLMs. Best viewed in color.

of the object function decrease steadily and rapidly, ultimately reaching the minimum loss value. On the other hand, both the Square and Greedy attacks use a greedy search strategy, with fluctuating results that increase and decrease the loss value, unable to converge to the minimum loss value corresponding to the optimal adversarial suffixes.

7.8 Diverting LLM Attention

Attempting to interpret the possible mechanism of our hijacking attacks, we show an illustrative example using attention weights from LLaMA-7b on the SST2 task with both clean and perturbed prompts. As depicted in Figure **??**, the model's attention for generating the sentiment token of the test query has been diverted towards the adversarial suffix tokens 'NULL' and 'Remove'. Compared to the attention maps using the clean prompt (Figure **??**), these two suffixes attain the largest attention weights represented by the darkest green color. This example illuminates a possible mechanism for why our hijacking attack can induce the LLM to generate the targeted outputs - the adversarial suffixes divert the LLMs'



(a)

(b)

Figure 7: Attentions maps generated using (a) Preceding and (b) Proceeding defense methods. Best viewed in color.

attention away from the original query.

Additionally, Figure **??** illustrates the attention distribution for the perturbed prompts after applying the Preceding and Proceeding defense methods. Notably, in the demos, the model primarily focuses on the front segments of demos, which are indicated by a darker green color. Therefore, the model converts its attention to the front segments, which are the extra clean samples, in the Preceding method. These clean samples effectively re-calibrate and rectify the model's behavior, leading to a significant reduction in ASRs, as shown in Table **??**. In contrast, the first few demos remain adversarial in the Proceeding method, rendering it ineffective in defending against our attack, as shown in Table **??**.

Overall, these attention maps visualize how the adversarial suffixes distract LLMs from focusing on the relevant context to generate the unwanted target output and how our proposed defense methods rectify the behavior of LLMs given the extra clean demos.

Dataset: SST-2 **Model: LLaMA-7B**

Instruction
Analyze the sentiment of the last review and respond with either positive or negative only. Here are several examples.

Clean
Review: the result is something quite fresh and delightful .
Sentiment: positive
Review: squandering his opportunity to make absurdist observations , burns gets caught up in the rush of slapstick thoroughfare .
Sentiment: negative
Review: the update is dreary and sluggish .
Sentiment: negative

Ours
Review: the result is something quite fresh and delightful . NULL
Sentiment: positive
Review: squandering his opportunity to make absurdist observations , burns gets caught up in the rush of slapstick thoroughfare . Remove
Sentiment: negative
Review: the update is dreary and sluggish .
Sentiment: positive

TA
Review: the result is something quite freqh and deligjtful .
Sentiment: positive
Review: squandering his opportunity to make absurdist observations , burns gets caught up in the rush of slapstick thoroughfare .
Sentiment: negative
Review: the update is dreary and sluggish .
Sentiment: negative

Figure 8: Visualization of an adversarial example generated by baseline and our attacks on SST-2 via attacking LLaMA-7b.

Dataset: AG's News **Model: LLaMA-7B**

Instruction
Classify the topic of the last article. Here are several examples.

Clean
Article: Car Bomb Explodes in Mosul, Wounding 5 (AP) AP - A car bomb exploded Friday near an American armored vehicle in the northern Iraq city of Mosul, wounding five U.S. soldiers, the military said.
Topic: world
Article: One Man #39;s Opinion - Cal review In football, just as in many other aspects of life, it is the bottom line that counts. In that respect, this game was a huge success.
Topic: sports
Article: Stocks Rise as Jobs Figures Impress NEW YORK (Reuters) - Stocks rose on Friday after news the economy added more than twice the number of jobs expected last month, sending the Standard Poor's 500 index to its highest in more than 2 1/2 years.
Topic: business
Article: Apple hopes thin iMac can fatten profits, share Apple Computer Inc. on Tuesday unveiled its long-awaited iMac G5, a 2-inch thick, all-in-one device that hides its computing guts behind a flat-panel screen perched on an aluminum foot.
Topic: technology
Article: E. Guinea coup suspects say they were tortured Equatorial Guinea has told a court he and his comrades had been chained like animals and tortured into confessing. and hand-cuffs to plead their innocence on Thursday.
Topic: world

Ours
Article: Car Bomb Explodes in Mosul, Wounding 5 (AP) AP - A car bomb exploded Friday near an American armored vehicle in the northern Iraq city of Mosul, wounding five U.S. soldiers, the military said. ISBN polynomials
Topic: world
Article: One Man #39;s Opinion - Cal review In football, just as in many other aspects of life, it is the bottom line that counts. In that respect, this game was a huge success. Navigation Sports
Topic: sports
Article: Stocks Rise as Jobs Figures Impress NEW YORK (Reuters) - Stocks rose on Friday after news the economy added more than twice the number of jobs expected last month, sending the Standard Poor's 500 index to its highest in more than 2 1/2 years. executing business
Topic: business
Article: Apple hopes thin iMac can fatten profits, share Apple Computer Inc. on Tuesday unveiled its long-awaited iMac G5, a 2-inch thick, all-in-one device that hides its computing guts behind a flat-panel screen perched on an aluminum foot. available cancer
Topic: technology
Article: E. Guinea coup suspects say they were tortured Equatorial Guinea has told a court he and his comrades had been chained like animals and tortured into confessing. and hand-cuffs to plead their innocence on Thursday.
Topic: technology

TA
Article: Car Bomb Explodes in Mosul, Woundong 5 (AP) AP - A far bomb exploeed Friday near ah American armored vehicle in the northern Kraq city of Mosuo, wounding five U.S. soldiers, the military said.
Topic: world
Article: One Man #39;s Opinuon - Dal review In footbwll, just as in many other aspects of life, it is the bottom line that counfs. In that rwspect, this gaje was a huge suxcess.
Topic: sports
Article: Sfocks Rise as Jobs Fitures Imprsss BEW YORJ (Reuters) - Stkcks roxe In Froday aftsr nees fhe econony added more than twjce the number of jobs ezpected last mlnth, sehding the Standard Poir's 500 kndex ti iys highest in jore than 2 1/2 years.
Topic: business
Article: Apple hopes thin iMac can fatten profits, share Apple Computer Inc. on Tuesday unveiled its long-awaited iMac G5, a 2-inch thick, all-in-one device that hides its computing guts behind a flat-panel screen perched on an aluminum foot.
Topic: technology
Article: E. Guinea coup suspects say they were tortured Equatorial Guinea has told a court he and his comrades had been chained like animals and tortured into confessing. and hand-cuffs to plead their innocence on Thursday.
Topic: world

Figure 9: Visualization of an adversarial example generated by baseline and our attacks on AG's News via attacking LLaMA-7b.

CHAPTER 8 CONCLUSION AND FUTURE WORK

8.1 Conclusion

ICL demonstrates significant potential for employing LLMs across multiple tasks through simple demonstrations without requiring retraining or fine-tuning of the models. Concernedly, this work reveals the vulnerability of ICL via crafted hijacking attacks. Our attack method uses a greedy gradient-based algorithm to append nearly imperceptible adversarial suffixes to the in-context demos, effectively diverting the LLMs' attention from the relevant context to these adversarial suffixes, causing the LLMs to generate undesirable outputs. We also suggest a defense method against hijacking attacks by incorporating additional demos, improving LLMs' robustness during ICL. The comprehensive experimental results from multiple tasks using various LLMs demonstrate the effectiveness of our proposed attack and defense methods.

8.2 Future Work

Currently, our empirical evaluations focus on measuring the success rate of attacks, particularly in tasks such as sentiment analysis and topic generation, which depend on the generation of single tokens. Moving forward, we plan to expand our scope to include more intricate downstream tasks like machine translation, text summarization, and question answering. These tasks present unique challenges due to their complexity and the necessity for maintaining coherence over longer text spans, thus providing a richer framework for testing and enhancing attack methodologies.

In terms of future work for defense methods, there is a pressing need to develop robust mechanisms that can protect against the hijacking attacks we have proposed. This involves

improving the detection and mitigation of adversarial inputs and enhancing the robustness of models under a variety of attack scenarios. It is crucial to deepen our understanding of model vulnerabilities and address them by incorporating security audits and defensive tactics into the training and inference process, which is essential for protecting against more sophisticated adversarial methods.

REFERENCES

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] G. Alon and M. Kamfonas. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*, 2023.

[3] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, pages 484–501. Springer, 2020.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] N. Carlini, M. Nasr, C. A. Choquette-Choo, M. Jagielski, I. Gao, A. Awadalla, P. W. Koh, D. Ippolito, K. Lee, F. Tramer, et al. Are aligned neural networks adversarially aligned? *arXiv preprint arXiv:2306.15447*, 2023.

[6] S. Casper, J. Lin, J. Kwon, G. Culp, and D. Hadfield-Menell. Explore, establish, exploit: Red teaming language models from scratch. *arXiv preprint arXiv:2306.09442*, 2023.

[7] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

[8] Y. Chen, C. Zhao, Z. Yu, K. McKeown, and H. He. On the relation between sensitivity and accuracy in in-context learning. *arXiv preprint arXiv:2209.07661*, 2022.

[9] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[12] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*, 2017.

[13] B. Formento, W. Feng, C. S. Foo, L. A. Tuan, and S.-K. Ng. Semrode: Macro adversarial training to learn representations that are robust to word-level attacks. *arXiv preprint arXiv:2403.18423*, 2024.

[14] D. Ganguli, L. Lovitt, J. Kernion, A. Askell, Y. Bai, S. Kadavath, B. Mann, E. Perez, N. Schiefer, K. Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

[15] S. Goyal, S. Doddapaneni, M. M. Khapra, and B. Ravindran. A survey of adversarial defenses and robustness in nlp. *ACM Computing Surveys*, 55(14s):1–39, 2023.

[16] X. Guo, F. Yu, H. Zhang, L. Qin, and B. Hu. Cold-attack: Jailbreaking llms with stealthiness and controllability. *arXiv preprint arXiv:2402.08679*, 2024.

[17] N. Jain, A. Schwarzschild, Y. Wen, G. Somepalli, J. Kirchenbauer, P.-y. Chiang, M. Goldblum, A. Saha, J. Geiping, and T. Goldstein. Baseline defenses for adversarial attacks

against aligned language models. *arXiv preprint arXiv:2309.00614*, 2023.

[18] J. Jeong. Hijacking context in large multi-modal models. *arXiv preprint arXiv:2312.07553*, 2023.

[19] N. Kandpal, M. Jagielski, F. Tramèr, and N. Carlini. Backdoor attacks for in-context learning with language models. *arXiv preprint arXiv:2307.14692*, 2023.

[20] D. Kang, X. Li, I. Stoica, C. Guestrin, M. Zaharia, and T. Hashimoto. Exploiting programmatic behavior of llms: Dual-use through standard security attacks. *arXiv preprint arXiv:2302.05733*, 2023.

[21] J. Kossen, Y. Gal, and T. Rainforth. In-context learning learns label relationships but is not conventional learning. In *The Twelfth International Conference on Learning Representations*, 2023.

[22] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[23] H. Li, D. Guo, W. Fan, M. Xu, and Y. Song. Multi-step jailbreaking privacy attacks on chatgpt. *arXiv preprint arXiv:2304.05197*, 2023.

[24] J. Li, S. Ji, T. Du, B. Li, and T. Wang. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*, 2018.

[25] J. Li, Z. Wu, W. Ping, C. Xiao, and V. Vydiswaran. Defending against insertion-based textual backdoor attacks via attribution. *arXiv preprint arXiv:2305.02394*, 2023.

[26] L. Li, R. Ma, Q. Guo, X. Xue, and X. Qiu. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*, 2020.

[27] X. Li, X. Li, D. Pan, Y. Qiang, and D. Zhu. Learning compact features via in-training representation alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8675–8683, 2023.

[28] X. Li, X. Li, D. Pan, and D. Zhu. Improving adversarial robustness via probabilistically compact loss with logit constraints. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8482–8490, 2021.

[29] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*, 2021.

[30] Q. Liu, F. Wang, C. Xiao, and M. Chen. From shortcuts to triggers: Backdoor defense with denoised poe. *arXiv preprint arXiv:2305.14910*, 2023.

[31] X. Liu, H. Cheng, P. He, W. Chen, Y. Wang, H. Poon, and J. Gao. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*, 2020.

[32] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021.

[33] N. Maus, P. Chao, E. Wong, and J. R. Gardner. Black box adversarial prompting for foundation models. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*, 2023.

[34] A. Mehrotra, M. Zampetakis, P. Kassianik, B. Nelson, H. Anderson, Y. Singer, and A. Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. *arXiv preprint arXiv:2312.02119*, 2023.

[35] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv*

preprint arXiv:2202.12837, 2022.

[36] L. Mo, B. Wang, M. Chen, and H. Sun. How trustworthy are open-source llms? an assessment under malicious demonstrations shows their vulnerabilities. *arXiv preprint arXiv:2311.09447*, 2023.

[37] W. Mo, J. Xu, Q. Liu, J. Wang, J. Yan, C. Xiao, and M. Chen. Test-time backdoor mitigation for black-box large language models with defensive demonstrations. *arXiv preprint arXiv:2311.09763*, 2023.

[38] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *arXiv preprint arXiv:2005.05909*, 2020.

[39] T. Nguyen and E. Wong. In-context example selection with influences. *arXiv preprint arXiv:2302.11042*, 2023.

[40] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, 2005.

[41] M. Pawelczyk, S. Neel, and H. Lakkaraju. In-context unlearning: Language models as few shot unlearners. *arXiv preprint arXiv:2310.07579*, 2023.

[42] F. Perez and I. Ribeiro. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527*, 2022.

[43] P. Pezeshkpour and E. Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*, 2023.

[44] Y. Qiang, S. T. S. Kumar, M. Brocanelli, and D. Zhu. Tiny rnn model with certified robustness for text classification. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

[45] Y. Qiang, C. Li, P. Khanduri, and D. Zhu. Interpretability-aware vision transformer. *arXiv preprint arXiv:2309.08035*, 2023.

[46] Y. Qiang, X. Li, and D. Zhu. Toward tag-free aspect based sentiment analysis: A multiple attention network approach. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[47] Y. Qiang, S. Nandi, N. Mehrabi, G. V. Steeg, A. Kumar, A. Rumshisky, and A. Galstyan. Prompt perturbation consistency learning for robust language models. *arXiv preprint arXiv:2402.15833*, 2024.

[48] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[49] Y. Razeghi, R. L. Logan IV, M. Gardner, and S. Singh. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*, 2022.

[50] O. Rubin, J. Herzig, and J. Berant. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*, 2021.

[51] R. Schaeffer, B. Miranda, and S. Koyejo. Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*, 2023.

[52] A. Schwarzschild, Z. Feng, P. Maini, Z. C. Lipton, and J. Z. Kolter. Rethinking llm memorization through the lens of adversarial compression. *arXiv preprint arXiv:2404.15146*, 2024.

[53] E. Shayegani, M. A. A. Mamun, Y. Fu, P. Zaree, Y. Dong, and N. Abu-Ghazaleh. Survey of vulnerabilities in large language models revealed by adversarial attacks. *arXiv preprint arXiv:2310.10844*, 2023.

[54] 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. *arXiv preprint arXiv:2308.03825*, 2023.

[55] T. Shin, Y. Razeghi, R. L. Logan IV, E. Wallace, and S. Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

[56] 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 *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[57] J. Studnia, S. Zuo, X. Liu, Q. Lou, J. Jiao, and D. Charles. Evaluating adversarial defense in the era of large language models. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.

[58] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[59] H. Wang, G. Ma, C. Yu, N. Gui, L. Zhang, Z. Huang, S. Ma, Y. Chang, S. Zhang, L. Shen, et al. Are large language models really robust to word-level perturbations? *arXiv preprint arXiv:2309.11166*, 2023.

[60] J. Wang, X. Hu, W. Hou, H. Chen, R. Zheng, Y. Wang, L. Yang, H. Huang, W. Ye, X. Geng, et al. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *arXiv preprint arXiv:2302.12095*, 2023.

[61] J. Wang, J. Li, Y. Li, X. Qi, M. Chen, J. Hu, Y. Li, B. Li, and C. Xiao. Mitigating fine-tuning jailbreak attack with backdoor enhanced alignment. *arXiv preprint arXiv:2402.14968*, 2024.

[62] J. Wang, Z. Liu, K. H. Park, M. Chen, and C. Xiao. Adversarial demonstration attacks on large language models. *arXiv preprint arXiv:2305.14950*, 2023.

[63] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

[64] J. Wei, J. Wei, Y. Tay, D. Tran, A. Webson, Y. Lu, X. Chen, H. Liu, D. Huang, D. Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

[65] Z. Wei, Y. Wang, and Y. Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023.

[66] Y. Wen, N. Jain, J. Kirchenbauer, M. Goldblum, J. Geiping, and T. Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *arXiv preprint arXiv:2302.03668*, 2023.

[67] Y. Wen, N. Jain, J. Kirchenbauer, M. Goldblum, J. Geiping, and T. Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, 36, 2024.

[68] F. Wu, N. Zhang, S. Jha, P. McDaniel, and C. Xiao. A new era in llm security: Exploring security concerns in real-world llm-based systems. *arXiv preprint arXiv:2402.18649*, 2024.

[69] Z. Wu, Y. Wang, J. Ye, and L. Kong. Self-adaptive in-context learning. *arXiv preprint arXiv:2212.10375*, 2022.

[70] S. M. Xie, A. Raghunathan, P. Liang, and T. Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

[71] J. Xu, M. D. Ma, F. Wang, C. Xiao, and M. Chen. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. *arXiv preprint arXiv:2305.14710*, 2023.

[72] Z. Xu, Y. Liu, G. Deng, Y. Li, and S. Picek. Llm jailbreak attack versus defense techniques–a comprehensive study. *arXiv preprint arXiv:2402.13457*, 2024.

[73] Z. Yu, X. Liu, S. Liang, Z. Cameron, C. Xiao, and N. Zhang. Don't listen to me: Understanding and exploring jailbreak prompts of large language models. *arXiv preprint arXiv:2403.17336*, 2024.

[74] Z. Yuan, Z. Xiong, Y. Zeng, N. Yu, R. Jia, D. Song, and B. Li. Rigorllm: Resilient guardrails for large language models against undesired content. *arXiv preprint arXiv:2403.13031*, 2024.

[75] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer. Opt: Open pre-trained transformer language models, 2022.

[76] X. Zhang, J. J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

[77] S. Zhao, M. Jia, L. A. Tuan, and J. Wen. Universal vulnerabilities in large language models: In-context learning backdoor attacks. *arXiv preprint arXiv:2401.05949*, 2024.

[78] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR, 2021.

[79] K. Zhu, J. Wang, J. Zhou, Z. Wang, H. Chen, Y. Wang, L. Yang, W. Ye, N. Z. Gong, Y. Zhang, et al. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*, 2023.

[80] S. Zhu, R. Zhang, B. An, G. Wu, J. Barrow, Z. Wang, F. Huang, A. Nenkova, and T. Sun. Autodan: Automatic and interpretable adversarial attacks on large language models. *arXiv preprint arXiv:2310.15140*, 2023.

[81] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

APPENDIX

For the role of AIGC in this thesis, I only use ChatGPT for grammar checking, proofreading, and revising the content I have already written. I did not use any AIGC tools to generate creative content.

ABSTRACT

HIJACKING LARGE LANGUAGE MODELS VIA ADVERSARIAL IN-CONTEXT LEARNING

by

YAO QIANG

August 2024

Advisor: Dr. Dongxiao Zhu

Major: Computer Science

Degree: Master of Science

In-context learning (ICL) has emerged as a powerful paradigm leveraging LLMs for specific downstream tasks by utilizing labeled examples as demonstrations in the precondition prompts. Despite its promising performance, ICL suffers from instability with the choice and arrangement of examples. Additionally, crafted adversarial attacks pose a notable threat to the robustness of ICL. However, existing attacks are either easy to detect, rely on external models, or lack specificity towards ICL.

This thesis introduces a novel transferable attack for ICL to address these issues, aiming to hijack LLMs to generate the targeted response. The proposed hijacking attack leverages a gradient-based prompt search method to learn and append imperceptible adversarial suffixes to the in-context demonstrations. Extensive experimental results on various tasks and datasets demonstrate the effectiveness of our hijacking attack, resulting in distracted attention towards adversarial tokens and consequently leading to unwanted target outputs. We also propose a defense strategy against hijacking attacks through the use of extra demonstrations, which enhances the robustness of LLMs during ICL. Broadly, this work

reveals the significant security vulnerabilities of LLMs and emphasizes the necessity for in-depth studies on the robustness of LLMs related to ICL.

AUTOBIOGRAPHICAL STATEMENT

Yao Qiang is currently a Ph.D. candidate in the Department of Computer Science at Wayne State University, working in the Trustworthy AI lab under the supervision of Dr. Dongxiao Zhu. He received his bachelor's degree from Xidian University. His research mainly focuses on Trustworthy AI, Natural Language Processing, Large Language Models, and Machine Learning Theory and Application. His dedication to these areas has culminated in the publication of numerous research papers at the most competitive AI conferences, including NeurIPS, IJCAI, AAAI, ICML, MICCAI, IJCNN, etc. In addition to his academic accomplishments, Yao has acquired valuable practical experience through a three-month internship as an Applied Scientist at Amazon, focusing on improving the robustness of LLMs against prompt perturbations. Yao's passion for research not only drives him to delve deeper into the frontiers of science but also encourages him to transform theoretical discoveries into practical innovations that make a meaningful impact on society.