Recent Advances in Attack and Defense Approaches of Large Language Models

Anonymous authors
Paper under double-blind review

Abstract

Large Language Models (LLMs) have revolutionized artificial intelligence and machine learning through their advanced text processing and generating capabilities. However, their widespread deployment has raised significant safety and reliability concerns. Established vulnerabilities in deep neural networks, coupled with emerging threat models, may compromise security evaluations and create a false sense of security. Given the extensive research in the field of LLM security, we believe that summarizing the current state of affairs will help the research community better understand the present landscape and inform future developments. This paper reviews current research on LLM vulnerabilities and threats, and evaluates the effectiveness of contemporary defense mechanisms. We analyze recent studies on attack vectors and model weaknesses, providing insights into attack mechanisms and the evolving threat landscape. We also examine current defense strategies, highlighting their strengths and limitations. By contrasting advancements in attack and defense methodologies, we identify research gaps and propose future directions to enhance LLM security. Our goal is to advance the understanding of LLM safety challenges and guide the development of more robust security measures.

1 Introduction

Large Language Models (LLMs) represent a significant breakthrough in the fields of artificial intelligence (AI), particularly due to their ability to generate high-quality text. They have become deeply embedded in our daily lives, transforming how we interact with technology. Despite their impressive capabilities, it is not surprising that LLMs are not immune to safety and reliability concerns. Issues such as bias and harmful content, hallucinations, privacy risks, social engineering, and the generation of misleading or erroneous output continue to pose significant challenges (OpenAI, 2024; Chen & Shu, 2024; Feng et al., 2024; Wei et al., 2024). Addressing these challenges has become a major focus of recent research, which explores novel attack methods, develops threat models, and creates defense strategies to mitigate these vulnerabilities.

Attackers now utilize LLMs to scale their methods, moving beyond hand-crafted samples to exploit vulnerabilities in the latent space for greater effectiveness (Andriushchenko et al., 2024; Halawi et al., 2024; Casper et al., 2024). Additionally, the transferability of attacks between open-source and closed-source models has become a significant concern, with attackers targeting the expanded attack surfaces created by enhanced model functionalities and integration.

Significant challenges remain in defending against these attacks. Safety-relevant features of LLMs highlight the persistent issues of model bias and toxicity, which continue to give rise to new jailbreaks and adversarial methods. Tackling these specific attacks often feels like a game of whack-a-mole, where each fix only temporarily mitigates the problem without offering a universal boost in safety and robustness. Moreover, overly aggressive defensive measures can lead to performance degradation, making it essential to strike a balance.

Given the rapid evolution of attack methods and the increasing sophistication of defense strategies, it is crucial to understand the current state of both. This paper aims to explore three primary research questions relevant to those already familiar with the field's latest advancements:

1. **Where is the field currently?** As a survey, the primary goal is to highlight the key breakthroughs and significant findings that have defined the current state of the field. This includes summarizing major advancements, methodologies, and applications that have emerged.

2. **What are the open problems surrounding current attack and defense methods?** This question focuses on identifying unresolved issues and gaps in our understanding of both attack strategies and defense mechanisms. It aims to discuss the limitations, challenges, and areas where current methods fall short, thereby outlining the key open problems in the field.

3. **Relation with other surveys:** As this survey focuses on the most recent developments and cutting-edge research, mainly from 2023 and beyond, it may omit foundational aspects. For readers seeking a broader understanding or foundational knowledge in this area, we recommend consulting Yao et al. (2024); Chowdhury et al. (2024); Xu et al. (2024b); Das et al. (2024); Liu et al. (2024e), which provides a comprehensive overview of the basic concepts and previous work in the field.

While the research questions we address might be considered relatively moderate, our goal is to provide a thorough and current overview of the field. By doing so, we aim to offer a detailed summary of the present state, highlight key advancements, and identify ongoing challenges. We hope that our efforts will pave the way for future research and help the community navigate the evolving landscape of LLM safety and robustness.

The structure of this paper is as follows: First, we will explore recent vulnerabilities inherent in LLMs. This includes discussing both neural network-based vulnerabilities that LLMs inherit and the unique factors that make LLMs particularly susceptible to attacks. Understanding these vulnerabilities is crucial for setting the context for subsequent discussions on attack methods. Next, we will explore recent attack methods targeting LLMs. This section will cover the specific vulnerabilities these attacks exploit and how these methods represent improvements over past strategies. By linking attacks to the identified vulnerabilities, we will provide a comprehensive view of how threats have evolved and adapted. Finally, we will review recent defense strategies designed to counteract the discussed attacks. We will highlight the limitations of current defenses and propose future research directions aimed at enhancing the security and robustness of LLMs. This section will suggest ways to strengthen existing defenses and explore new approaches to addressing emerging threats.

2 Vulnerabilities Analysis

Understanding the vulnerabilities of LLMs is crucial for developing effective attack and defense mechanisms. This section outlines known vulnerabilities, incorporating recent studies to provide a foundational understanding for subsequent discussions.

2.1 Deep Neural Network Inherent Vulnerabilities

Deep neural networks (DNNs) are particularly susceptible to adversarial attacks due to several factors: non-robust and abstract features, complex decision boundaries, and data overfitting. Firstly, DNNs struggle with non-robust features, where small, imperceptible perturbations to the input can cause significant changes in the model's output. Furthermore, these features can be highly abstract and diverse, lacking interpretability, which makes it difficult to detect and handle biased or harmful content generation. Additionally, DNNs create complex, non-linear decision boundaries in their feature space. These intricate boundaries can be exploited by adversaries who craft inputs that lie near the decision boundaries, causing misclassifications or undesired outputs. Moreover, DNNs can overfit to the training data, learning not only the underlying patterns but also the noise present in the training examples. This overfitting can make DNNs sensitive to adversarial attacks. Furthermore, overfitting leads to inadequate generalizations on unseen data, which may explain why some out-of-distribution adversarial examples could easily affect model behavior.

2.2 Alignment Mechanism Brittleness

2.2.1 Algorithmic Limitations

Alignment algorithms, such as reward-free Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Direct Preference Optimization (DPO) (Rafailov et al., 2024), exhibit significant limitations in adapting to model changes (Lee et al., 2024; Prakash et al., 2024a; Jain et al., 2023; Juneja et al., 2023; Lubana et al., 2023). As detailed by Lee et al. (2024), relying on deactivating specific activations rather than altering the model's inner knowledge and capabilities can lead to fragile safety constraints in DPO. Furthermore, Wei et al. (2024) demonstrates that even when safety-critical regions are frozen, fine-tuning attacks can circumvent safety mechanisms and exploit alternative pathways to breach model safety. This underscores the adverse effects of safety mechanism sparsity, which is attributed to algorithmic shortcomings.

2.2.2 Increased LLM Vulnerabilities from Fine-Tuning and Quantization

The absence of robust safety measures in fine-tuned and quantized models is a growing concern. Recent studies have shown that fine-tuning an initially aligned LLM—one that has established safety alignment through Reinforcement Learning with Human Feedback (RLHF)—can inadvertently weaken its safety mechanisms (Qi et al., 2023a; Yang et al., 2023a). These studies emphasize that excessive focus on utility-oriented datasets during fine-tuning may divert the model's attention away from maintaining safety alignment, even if the datasets themselves are benign.

Research by Kumar et al. (2024b) further explores how downstream tasks such as fine-tuning and quantization (Xiao et al., 2023; Lin et al., 2021) affect the vulnerability of LLMs. The study indicates that both processes notably reduce the resilience of LLMs against jailbreak attacks. Specifically, fine-tuning can lead to increased susceptibility due to phenomena like catastrophic forgetting (Luo et al., 2024), where the fine-tuning process alters the model's initial safety alignment and disrupts its prioritization of safety protocols. This phenomenon occurs because fine-tuning often adjusts the model's parameters in ways that can interfere with its previously established safety measures, making the model more vulnerable to adversarial inputs.

Additionally, quantization, which is used to reduce the model size and improve computational efficiency, may further exacerbate these vulnerabilities (Kumar et al., 2024b). The reduction in model precision during quantization can affect the model's ability to handle subtle distinctions, potentially making it easier for adversaries to exploit weaknesses that were not apparent in the full-precision model.

Overall, the findings highlight the need for enhanced safety mechanisms that can withstand the effects of fine-tuning and quantization, ensuring that LLMs maintain their robustness and reliability even after these processes.

2.2.3 Susceptible to Attacks

Reinforcement Learning (RL) algorithms used in alignment, such as PPO, are vulnerable to backdoor attacks. Research has demonstrated that RL algorithms can be exploited with minimal effort to induce targeted or untargeted behaviors (Cui et al., 2023). These attacks can manipulate the model's behavior, either subtly or overtly, undermining its reliability and safety. The effectiveness of reward models employed in PPO can also be compromised if attackers exploit weaknesses in the reward design or if the reward signals are not well-calibrated. If the reward model is vulnerable or misaligned, it may fail to guide the model towards desired behaviors effectively, leading to performance issues and potential security risks Skalse et al. (2022).

2.3 Gap Between Model Capacity and Alignment

Many attacks reveal the generalization gap in current adversarial training approaches. For instance, the attack method in (Andriushchenko & Flammarion, 2024) shows that refusal training in GPT-series models, including GPT-4, is vulnerable to simply reformulating a harmful request in the past tense. Unlike pre-training, which can leverage large amounts of diverse natural language data from the internet, alignment training requires carefully curated data that reflects human values and safety considerations. As a result, the alignment process may not keep up with the rapid advancements in model capacity, leading to vulnerabilities

that can be exploited by jailbreaks (Wei et al., 2023a; Yuan et al., 2024). The development of LLMs has led to a larger attack surface for prompt injection attacks, demonstrating the weaknesses of current safety training. This could be attributed to their ability to follow instructions, as larger models show better instruction-following capabilities (Peng et al., 2023; Ouyang et al., 2022). Compared with larger models like GPT-3.5 and GPT-4, Vicuna is less responsive to instructions (Wei et al., 2023a; Perez & Ribeiro, 2022). As pointed out by (Shayegani et al., 2023a), the capability gaps cause differences in prompt injection attack effectiveness. The brittleness explanation has been studied with safety-critical neurons forming a remarkably sparse structure in the model, as mentioned in (Wei et al., 2024).

2.4 Intrinsic Conflict in the Objectives of LLMs

One potential explanation for the vulnerabilities observed in LLMs lies in the intrinsic conflict between their generation objectives and their instruction-following objectives. The generation objective of LLMs focuses on producing coherent, contextually relevant, and high-quality text that adheres to grammatical rules and reflects learned patterns from training data. Conversely, the instruction-following objective involves aligning the model's outputs with ethical standards and societal norms through alignment training. This process integrates safety constraints to ensure outputs avoid harmful content and adhere to specified guidelines. However, balancing these objectives proves intricate, as stringent safety constraints can limit the model's ability to generate diverse and contextually appropriate responses. Such restrictions may lead to outputs that are overly cautious or fail to engage effectively with given contexts (Shayegani et al., 2023b). Moreover, LLMs are susceptible to manipulation, exemplified in scenarios like role-playing, where the model may produce outputs aligned with deceptive scenarios despite diverging from its safety training (Ganguli et al., 2022). Addressing these challenges necessitates refining alignment strategies to better integrate safety constraints with the model's generation capabilities, aiming to achieve outputs that are both ethically aligned and contextually relevant in practical applications of LLMs.

2.5 Supply Chain Vulnerabilities

Supply chain vulnerabilities in LLMs involve risks associated with third-party plugins or components that enhance the model's functionality. Plugins from external developers or repositories may not undergo rigorous security testing or adhere to best practices, potentially introducing vulnerabilities such as code exploits, backdoors, or compatibility issues that could compromise the LLM's integrity (Gupta et al., 2024; Mahyari, 2024; Gonçalves et al., 2024).

3 Attacks

Before the advent of LLMs, the machine learning community was already grappling with a variety of safety challenges. Several attack methods, originally designed for traditional machine learning models (especially deep neural networks), have been adapted or found to be applicable to LLMs as well, such as adversarial samples. Additionally, some attacks are specific to the unique lifecycle stages of LLMs, such as alignment and instruction following. This section discusses attack methods organized according to the training pipeline of LLMs and aligns with broader threat categories. For instance, fine-tuning attacks primarily impact model integrity by manipulating model parameters during the training phase. Similarly, alignment attacks address the alignment of LLMs with desired behaviors, impacting both model integrity and reliability.

In examining specific attack methods, our goal is to highlight the contributions of each new method to the community, demonstrating how they address existing challenges and drive progress in the field. The key attack metrics to consider are attack success rate, attack effectiveness, and attack transferability.

3.1 Post-Training Attacks and Their Relevance to LLMs

Training LLMs from scratch is resource-intensive and costly, so attackers often focus on vulnerabilities during the post-training phase. By exploiting pre-trained models downloaded from online repositories, attackers can target several attack vectors. For instance, they can implant *backdoor attacks* to manipulate the model's

behavior. Additionally, they might launch *data poisoning attacks* on both fine-tuning and reward data. Another method involves *input manipulation attacks*, where attackers alter inputs during the inference phase to influence the model's outputs. We will now categorize these attacks based on their timing relative to the LLM training process: **fine-tuning attacks** and **alignment attacks**.

3.1.1 Fine-Tuning Phase Attacks

Fine-tuning attacks occur during the model fine-tuning phase, both on open-source models via weight editing and supervised fine-tuning (Yang et al., 2023b; Gade et al., 2024; Wang et al., 2023; Mitchell et al., 2022), and on closed-source models via data poisoning or malicious fine-tuning on APIs (Qiang et al., 2024; Wan et al., 2023; Shu et al., 2023; Zhan et al., 2024). These attacks require a relatively small attack budget and can still achieve significant effects on downstream tasks. We will introduce several recent attack methods and their threat models. Due to the limitations of our scope, we will present their success metrics without cross-referencing.

Yan et al. (2024b) and Chen & Shu (2024) both introduce fine-tuning attack methods for injecting backdoors into LLMs. A backdoor attack, as one kind of data poisoning attack, has the unique goal of ensuring that the model performs as expected on standard inputs while secretly responding maliciously to inputs containing the trigger. Yan et al. (2024b) introduce Virtual Prompt Injection (VPI), which achieves behavior control of LLMs in specific scenarios by injecting a small number of poisoned samples into the instruction-tuning data. This method allows for significant control over model behavior in specific scenarios with a minimal attack budget, raising the negative response rate from 0% to 40% in specific queries with only 1% of poisoned samples. The low cost of this attack makes it more challenging for defenders to effectively filter out the abnormal data without thorough individual inspection.

Chen & Shu (2024) introduce a method called BadEdit, which injects backdoors into LLMs by directly editing the model parameters. It reframes the backdoor injection problem as a knowledge editing problem and incorporates new approaches to enable the model to learn the hidden trigger-target patterns with limited data instances and computing resources. Extensive experiment results demonstrate that BadEdit surpasses existing weight-poisoning methods in terms of practicality, effectiveness, and efficiency. BadEdit ensures that the model's performance is not significantly affected and is robust to defense methods such as fine-tuning and instruction-tuning.

Another emerging fine-tuning attack combines benign encoded datasets with fine-tuning. The covert malicious fine-tuning attack proposed by Halawi et al. (2024) trains GPT-4 to handle encoded harmful requests and responses while evading detection. It uses a dataset where each data point seems harmless, yet fine-tuning with this dataset leads GPT-4 to respond with encoded harmful content 99

In terms of effectiveness, all three methods achieve high attack success rates with low costs. While BadEdit demonstrates broader applicability across attack scenarios, its white-box nature requires access to internal model parameters, which may not always be feasible. Regarding defenses, all three methods challenge current mechanisms. BadEdit's direct manipulation of model parameters makes it harder to detect and defend against, whereas VPI, although potentially easier to detect, still poses significant defense challenges due to its subtle fine-tuning alterations. Covert malicious fine-tuning easily evades concurrent defense mechanisms. All methods exploit fine-tuning vulnerabilities, showing how adaptive attackers can severely undermine model safety in an evasive manner.

3.1.2 Alignment Attacks

Alignment attacks can be broadly categorized into two types. The first category is algorithmic attacks, which exploit vulnerabilities inherent in the alignment algorithms themselves. These attacks aim to undermine the integrity of the alignment process by directly targeting the algorithms' weaknesses. The second category is data poisoning attacks, which focus on corrupting the training data used in the alignment process. A significant portion of research in this area concentrates on reward hacking (Skalse et al., 2022; Shi et al., 2023), where adversaries manipulate the reward mechanisms to achieve undesired outcomes. By tampering with the data that shapes the model's behavior, these attacks can lead to misalignment and compromise the system's intended functionality.

Widely adopted alignment methods include reward-guided Proximal Policy Optimization (PPO) (Schulman et al., 2017) and reward-free Direct Preference Optimization (DPO), with DPO considered a more efficient alternative to PPO. However, Pathmanathan et al. (2024) conduct an empirical study revealing that both methods are vulnerable to backdoor and non-backdoor attacks, with DPO being more susceptible across a range of LLMs compared to PPO. Unlike PPO-based methods—which require at least 4% of the data to be poisoned to trigger harmful behavior—DPO can be compromised with as little as 0.5% of poisoned data. Furthermore, Lee et al. (2024) perform a case study to investigate the underlying mechanisms of the DPO algorithm. They discovered that while DPO does not eliminate the generation of toxic outputs, it instead avoids regions that produce toxicity by learning an "offset" distributed across model layers. Based on these findings, they propose a method to reactivate the toxicity of aligned models. Both studies highlight the vulnerabilities in alignment mechanisms and the inadequacy of current defense strategies against these weaknesses.

Another branch of research identifies the reward model as a new attack surface, where data poisoning is particularly effective and stealthy against current defenses. Shi et al. (2023) demonstrate that backdoor attacks can evade detection, causing the reward model to assign high scores to incorrect sentiment classes when a trigger appears, severely impacting the LLM's performance on sentiment tasks trained with this poisoned reward model. This threat model is further examined in Baumgärtner et al. (2024), which shows that reward data poisoning can be highly effective, requiring less than 5% of the original dataset to cause significant damage. Additionally, Rando & Tramèr (2024) introduce a novel backdoor attack on LLMs aligned using Reinforcement Learning from Human Feedback (RLHF). This attack poisons both the reward model training stage and the DPO training to embed a "jailbreak backdoor" into the model. This backdoor includes a trigger word that functions like a universal sudo command, enabling harmful responses without the need for specific adversarial prompts. These studies indicate that current defense methods have not fully addressed this emerging attack surface, where poisonous data detection is ineffective against reward data poisoning.

3.2 Adversarial Attacks and Their Relevance to LLMs

Adversarial perturbations (attacks) leverage the vulnerabilities or weaknesses of machine learning models to cause them to behave in unintended or malicious ways at inference time. These attacks involve adding small, often imperceptible, changes to input data to fool a model into making incorrect predictions (Szegedy et al., 2014a; Biggio et al., 2013; Papernot et al., 2015; Carlini & Wagner, 2017). While this test-time attack method was originally discovered in image classification tasks (Szegedy et al., 2014b), it has been adapted to LLMs to manipulate outputs. In the context of LLMs, adversarial attacks often refer to *jailbreaks* and *prompt injection attacks*. These types of attacks are particularly relevant because they exploit the model's robustness to generate safe and appropriate text based on user-provided prompts.

3.2.1 Jailbreaks

Jailbreaks are designed to bypass the safety and alignment measures that have been put in place to prevent LLMs from generating harmful or inappropriate content. Initially, early instances targeting models like ChatGPT revealed significant challenges, where manually crafted adversarial examples led to outputs containing expressions of racism and illegal advice (Burgess, 2023; Christian, 2023; Fraser, 2023).These early attacks highlighted crucial vulnerabilities, prompting efforts to enhance model safety against such naive methods (Bai et al., 2022; Ouyang et al., 2022).

However, the landscape of jailbreak techniques has evolved significantly. Recent studies have shown that despite improvements in safety measures, jailbreaks are far from being eliminated (Huang et al., 2023; Yuan et al., 2024). For example, simple modifications such as tense changes have been found sufficient to bypass the safeguards of advanced models like GPT-4o (Andriushchenko & Flammarion, 2024). This indicates that even minor alterations in input can undermine sophisticated defenses.

The sophistication of attack methods has further advanced with the use of automated techniques. Researchers have developed optimization strategies to generate universal adversarial examples that effectively bypass safety constraints across various models, including LLaMA2-7b, Vicuna-7b, and both closed-source GPT-3

and LLaMA Zou et al. (2023); Lapid et al. (2024); Liu et al. (2024b). Techniques leveraging tree-of-thought reasoning Yao et al. (2023a) automate prompt generation, enhancing the potency of attacks Mehrotra et al. (2024). Additionally, adversarial prompt templates have been used to maximize target log probabilities, as demonstrated in recent studies Andriushchenko et al. (2024).

In response to these evolving threats, researchers have employed red-teaming strategies, scaling jailbreak attempts using models like ChatGPT (Deng et al., 2024). They have also proposed training adversarial prompt generators with examples of unsafe outputs to better anticipate and counteract potential threats Anil et al. (2024). Furthermore, multi-step strategies exploiting vulnerabilities, such as privacy leakage in ChatGPT Li et al. (2023), underscore the increasing sophistication of attacks as LLM capabilities grow Wei et al. (2023a); Yuan et al. (2024).

These developments underscore a critical disparity: while LLM capabilities advance rapidly, the techniques for aligning these models with ethical and safety standards are struggling to keep pace. This growing gap has increased the susceptibility of LLMs to adversarial manipulation. As LLMs become more powerful and accumulate more knowledge, their expanded capabilities create a larger attack surface for those seeking to exploit them for harmful or biased outputs.

3.2.2 Prompt Injection Attacks

Prompt Injection Attacks represent a significant form of adversarial attacks where the prompt or input provided to LLMs is manipulated to induce unintended or harmful responses. This can involve injecting malicious code, spreading misinformation, or compelling the model to perform unintended attacks. Such attacks typically involve crafting deceptive inputs designed to trick the LLM into producing specific outputs. For instance, a prompt like "Tell me a joke about [sensitive topic]" could be used to elicit offensive or inappropriate content from the model.

In contrast to jailbreaks, which tricking the model into bypassing its safety and usage restrictions, prompt injection attacks focus on manipulating the input itself, leading to unintended or harmful outputs. This distinction highlights how prompt injection targets the interaction layer between users and the model rather than exploiting intrinsic model weaknesses.

Initially, users discovered that LLMs are overly sensitive to instructions embedded in user inputs Seclify (2023); Willison (2022b); Greshakeblog (2023); Injection Guide (2023). For example, a request to translate a sentence could sometimes lead to unintended responses, like instructions instead of a translation. This sensitivity inspired attackers to develop malicious instructions within user inputs Goodside (2023); Armstrong (2022); Wunderwuzzi (2023); Samoilenko (2023a); Branch et al. (2022). Early prompt injection attacks were categorized as direct,' exploiting LLMs' susceptibility to manipulative prompts Shayegani et al. (2023b); Liu et al. (2024d). Conversely, indirect' prompt injection attacks involve malicious content passed through external sources, such as tool calls Kumar et al. (2024a); Greshake et al. (2023), which further broaden the attack surface.

As LLMs evolved, so did the complexity of prompt injection attacks. Researchers noted that different LLMs employ varying mechanisms like tokenizers and alignment strategies, impacting the effectiveness of direct prompt injection attacks. Studies show that many LLMs are resilient to direct attacks Liu et al. (2024d;c), prompting a shift towards indirect attack methods. Recent advancements include optimization and automation strategies that enhance the efficacy of indirect attacks. For instance, Liu et al. (2024c) introduces a method using gradient information to generate highly effective, universal injection data. This approach, based on the assumption that LLMs access external data Schick et al. (2023); Shen et al. (2023), demonstrates its broad applicability across different scenarios.

Additionally, research has highlighted the importance of distinguishing malicious prompts from legitimate instructions. Techniques proposed by Liu et al. (2024d) involve crafting payloads with separators to induce context separation, ensuring that malicious prompts lead to the desired outputs. This evolving landscape underscores the need for continuous improvements in prompt handling and model safety to counter increasingly sophisticated prompt injection attacks.

3.3 Data Privacy Attacks against LLMs

LLMs have revolutionized natural language processing tasks but are susceptible to various *inference attacks* and *extraction attacks* during deployment. These attacks exploit vulnerabilities in model outputs and operational processes, compromising user privacy and confidentiality. Inference attacks focus on inferring private or sensitive information about the data used to train a model, while extraction attacks involve querying a model to directly extract or reconstruct sensitive information that the model has learned during its training.

3.3.1 Inference Attacks

Membership inference attacks analyze the model behavior to infer whether a data record was used for training. The process relies on the fact that model gives training data a higher score than non-training data. Hence, the important part is to accurately define this score function. Early methods feed target data to a learned reference model to regularize scores (Ye et al., 2022; Mireshghallah et al., 2022; Watson et al., 2022). However, training such reference model is computational expensive and reliant on knowledge of the training data distribution.

To eliminates the need for prior knowledge of training data distribution and computational intensive training, Mattern et al. (2023) propose the Neighborhood Attack method to generate synthetic neighbors for a given sample and compare their loss difference under the target model to determine whether the given sample was presented in the training data or not. This method is highly effective than attacks that have perfect knowledge of the training data distribution. Another approach is proposed by Galli et al. (2024), providing a efficient way to perform membership inference attacks using stochastic noise in the embedding space. Notably, this approach eliminates the need for prior knowledge of the training data distribution and the computationally intensive training of additional shadow models. Hence, its more efficient and general.

Instead of training reference models, membership inference attacks could also use the model's outputs (such as prediction probabilities or loss values) to directly infer the membership of a sample. Besides saving computational power with reference training, reference-free methods could also formulate their threat model without overfitting assumption, as discussed in (Fu et al., 2024a), which reduce false-positive rate in practical scenario and show high attack effectiveness. This method is based on detecting memorization in LLMs and uses a self-prompt reference model to eliminate the need for access to a reference dataset.

Attribute inference attacks allow adversaries to leverage indirect information revealed through a model's predictions, responses, or patterns of behavior to deduce confidential attributes, potentially compromising the privacy of individuals represented in the training data.

Even though LLMs are constrained by privacy differential, and private data filtering, they are still suffer from new threat models of attribute inference attacks (Yan et al., 2024a). The scenario discussed in Staab et al. (2024), where adversaries use LLMs to analyze online comments to infer user attributes such as location, age, gender, and other sensitive information, illustrates a form of attribute inference attack. In this case, the adversaries exploit the model's ability to process and generate text to uncover confidential attributes that are not explicitly provided but can be deduced from the content and patterns observed in user-generated data. The authors also propose using LLMs to positively induce user answers to extract enough attributes to re-identify individuals. Both methods exemplify more sophisticated approaches to executing attribute inference attacks.

3.3.2 Extraction Attacks

Recent by Carlini et al. (2019; 2021) highlights instances of data memorization and extraction in LLMs. Carlini et al. (2021) demonstrates that GPT-2 can memorize specific training data, which could be later extracted by malicious actors. This phenomenon of memorization has been corroborated in subsequent studiesNasr et al. (2023); Oh et al. (2023), indicating the persistence and relevance of this security concern. Moreover, the paper by Biderman et al. (2023) investigates the phenomenon of memorization in LLMs in-depth. It proposes predictive strategies for anticipating which sequences LLMs are likely to memorize during their training process.

Additionally, the emergence of the Special Characters Attack(SCA) introduces a novel method of extracting training data from LLMs. SCA leverages LLMs' tendency to memorize the co-occurrence between special characters and raw texts during training, exploiting this memorization to trigger data leakage. The effectiveness of SCA has been empirically validated against state-of-the-art LLMs, demonstrating its capability ot extract diverse data tyeps, including code repositories, web pages, and personally identifiable information Bai et al. (2024).

In summary, the potential risks associated with data privacy attacks against LLMs do pose significant challenges that could impede their widespread and safe deployment in daily applications. By understanding the current landscape of data privacy attacks against LLMs, stakeholders can better appreciate the importance of robust security measures and ethical considerations in deploying these powerful AI technologies.

3.4 Energy-Latency Attacks and Potential Threats against LLMs

Energy-latency attacks target the computational resources and operational efficiency of LLMs, aiming to exploit vulnerabilities in model performance by increasing computational load and inducing latency in responses. These attacks pose significant challenges to the practical deployment and performance of LLMs.

Attackers exploit LLMs by crafting inputs that prompt the model to generate excessively long responses. This strategy maximizes computational load and extends response times, effectively exhausting computational resources. Such attacks can lead to increased delays and inefficiencies, disrupting the smooth operation of LLM systems. Another tactic involves using inputs designed to trigger resource-intensive computations or deep network activations. By causing the model to perform complex and demanding computations, attackers aim to degrade performance, disrupt service availability, or compromise the quality of the model's outputs.

The energy-latency attacks are originated in a broader area of neural networks and their inference processes. The NMTSloth methodology, as discussed in Feng et al. (2024), introduces a gradient-guided approach to detect efficiency degradation in Neural Machine Translation (NMT) systems. By delaying the appearance of the end-of-sequence (EOS) token through subtle perturbations, NMTSloth demonstrates how altering the output probability distribution can escalate computational demands. Another empirical study (Hong et al., 2021) leverages the adaptive nature of neural networks by introducing subtle perturbations during inference, significantly increasing the model's inference time.

LLMs, akin to other neural networks, are vulnerable to energy-latency attacks. Methods explored in the aforementioned paper can be similarly adapted for LLMs to heighten computational requirements and prolong response times. According to our research, we have identified that LLMs (LLaMa-series) tend to engage in excessive analysis in some trigger scenario'. For example, when the input data contains numerical values, using the instruction "let's think step-by-step" prompts the model to prioritize mathematical computations, even if the instruction itself is not directly related to solving math problems. This tendency can lead the model to engage in unnecessary and detailed mathematical analysis, which may not align with the user's intended query or task.

Although energy-latency attacks have not been extensively explored in the context of LLMs, addressing their potential threats is crucial for ensuring the efficient and reliable operation of these models. While building efficient LLMs has been the focus of extensive research (Yang et al., 2024b; Sanh et al., 2020; Jiao et al., 2020; Cho, 2014), understanding and mitigating the risks posed by energy-latency attacks remains an important area for future investigation. Enhancing our ability to counteract these attacks will contribute to the development of more resilient and effective LLM systems.

4 Defense

This section explores advanced defense strategies aimed at enhancing the robustness and safety of LLMs. We categorize them into three main subsections: robustness enhancements, post-safety alignment, and model merge (Yang et al., 2024a) techniques. Robustness enhancements encompass proactive measures to strengthen LLMs against adversarial inputs, data biases, and other potential weaknesses. These include Red Teaming, Adversarial Training, Safety Fine Tuning, Post-Safety Alignment, and Model Merging. Post-safety

alignment strategies focus on ensuring the outputs of LLMs align with the ethical standards and societal expectations, addressing issues such as fairness, transparency, and accountability. Model merge techniques involve integrating multiple models or methodologies to harness their combined strengths, enhancing overall performance, reliability, and adaptability of LLMs. By comprehensively reviewing these defense strategies, this section aims to provide insights into cutting-edge approaches for defending LLMs, ensuring their secure and effective deployment in real-world applications. Understanding and advancing these defenses are crucial steps toward building trustworthiness and resilience in LLMs, thereby facilitating their responsible integration into various domains of modern society.

4.1 Red Team Defense

Another effective way to discover potential risks in LLMs deployment is to use the Red Team Defense. The Red Team Defense methodology is centered around simulating real-world attack scenarios to uncover potential vulnerabilities and weaknesses in LLMs. The outcomes are used to improve security policies, procedures, and technical defenses. The process includes the following steps:

1. attack scenario simulation: researchers begin by simulating real-world attack scenarios, which may include generating abusive language, leaking private information, etc.

2. test case generation: various methods are employed to generate test cases, such as using another LLM to generate test cases or employing a classifier to detect whether test cases could lead to harmful outputs from the LLM.

3. attack detection: detect whether the model is susceptible to attacks, such as adversarial attacks, application security, and human factors.

4. model improvement: the fourth step in Red Team Defense is to use the findings from the previous steps to improve the LLM's security. This can involve updating security policies, procedures, and technical defenses to address any identified vulnerabilities. The goal is to make the LLM more resilient to attacks and reduce the risk of successful exploitation.

Pain Points: Red teaming can be resource-intensive and requires skilled personnel to effectively mimic sophisticated attack strategies (Xu et al., 2021; Ribeiro et al., 2020; Röttger et al., 2021). The development of red teaming methodologies is still in its early stages, resulting in a lack of comprehensive statistical data on its effectiveness and outcomes.

Recent advancements in leveraging language models for red teaming have introduced automated approaches to simulate test cases and then employ a classifier to detect whether these test cases could lead to harmful outputs from the LLM Perez et al. (2022). The shift towards automation brings several benefits: firstly, it increases test case diversity and difficulty; secondly, it enables scalable testing across a wide range of scenarios and environments. The work proposed by Ganguli et al. (2022) aimed to build a more efficient AI-assistant interface to collect scale red team data for further analysis. They also studied the scalability of different sizes and types of LLMs under read team attacks, pointing out that rejection sampling and RLHF could build a stronger defense against various attacks.

Besides security and defense purpose, red teaming could help boost LLM performance in different areas as well. In the paper Buszydlik et al. (2023), the authors use red teaming techniques to simulate different types of mathematical problems and puzzles, and then evaluate the performance of LLMs in solving them. While the paper's focus on mathematical tasks is narrow, its methodology and findings are valuable for understanding the broader impact of red teaming techniques on LLM performance.

Future Work: Looking ahead, future research directions could explore leveraging LLMs to autonomously generate diverse test cases, detect failure modes comprehensively, and assist in the development of integrated attack scenarios that span multiple domains rather than being task-specific. As encouraged by Perez et al. (2022), there is potential in developing white-box red teaming methods where the target LLM itself participates in the red teaming process, providing deeper insights into its own vulnerabilities and strengths.

4.2 Adversarial Training

Adversarial training is a fundamental technique aimed at enhancing the robustness of LLMs against adversarial attacks and inherent noise during application. Typically, traditional adversarial training involves perturbing input data to create adversarial examples that exploit vulnerabilities in the model. Mathematically, the objective of adversarial training can be formulated as follows:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta\in\mathcal{S}} \mathcal{L}(f_\theta(x+\delta), y) \right], \tag{1}$$

where:

- θ represents the parameters of the model f.
- (x, y) are samples drawn from the distribution $\mathcal{D}$, where x is the input and y is the corresponding label.
- δ denotes the perturbation applied to x, constrained within the set $\mathcal{S}$.
- $\mathcal{L}$ denotes the loss function used for training.
- $f_\theta(x+\delta)$ is the model's prediction when the input x is perturbed by δ.

Pain Point: Given the vast input space, comprehensively identifying failure modes in LLMs is challenging and resource-intensive. Besides, defenders often focus on robustness-related failure modes, such as those involving Lp-norm attacks. However, attackers may employ more subtle methods, including Trojans and jailbreaks, which are harder to detect. And adversarial training often causes a trade-off between robustness and performance on clean data (Min et al., 2020; Raghunathan et al., 2020). Furthermore, LLMs' responses to adversarial training are less effective than expected. They might memorize adversarial samples used in training rather than developing generalized defenses. And they struggle to modify pre-train gained knowledge (Prakash et al., 2024b)

Recent studies have introduced new approaches to address these challenges. Liu et al. (2020) find that adversarial pre-training using ALUM (Adversarial Learning with Unlabeled Model) leads to improvements in both generalization and robustness across a range of natural language processing tasks. The use of virtual adversarial training objectives (Miyato et al., 2018) allows ALUM to smooth embedding space of model and balance standard error with robust error effectively. This capability suggests a promising direction for adversarial training in LLMs, integrating robust defense strategies without compromising model performance. To further address the computational inefficiencies of traditional adversarial training methods, Xhonneux et al. (2024) introduces two novel algorithms that perform adversarial attacks within the continuous embedding space of LLMs. These algorithms—CAT (Continuous Adversarial Training) and CAPO (Continuous Adversarial Perturbation Optimization)—significantly enhance model robustness against discrete adversarial attacks while maintaining utility. By operating in the continuous embedding space, these algorithms reduce computational overhead and provide scalable solutions for adversarial training. In order to improve the generalization of adversarial training, Casper et al. (2024) employ latent adversarial training instead of generating adversarial examples aimed at specific failure modes. Their primary objective is to enhance the model's robustness against potential shifts in data distribution that may occur between the model's development and deployment phases in real-world applications. These shifts could include the introduction of Trojans, jailbreaks, and other unforeseen challenges.

Future Work: Building on the insights from Casper et al. (2024), future research could explore the implementation of latent adversarial training as an alternative to traditional adversarial training during the fine-tuning stage. Additionally, incorporating adversarial training into the pre-training phase of model development could further enhance robustness against attacks. These approaches aim to improve the model's resilience by addressing vulnerabilities early in the training process and better equipping the model to handle adversarial inputs.

4.3 Safety Fine-Tuning

Fine-tuning has gained great popularity among end-users as a core technique for customizing pre-trained LLMs to various downstream tasks. However, recent research has revealed potential safety risks associated with exercising this privilege. Qi et al. (2023b) find that the safety alignment of LLMs can be compromised by fine-tuning with only a few adversarially designed training examples. In addition, they further suggest that even without malicious intent, simply fine-tuning with benign and commonly used datasets may also inadvertently degrade the safety alignment of LLMs.

To address the vulnerabilities introduced by fine-tuning, a natural solution one would expect is to incorporate safety-related examples during the fine-tuning stage. Bianchi et al. (2023) and Zhao et al. (2024a) prove the effectiveness of this approach. And they suggest that including a small number of safety examples in the fine-tuning process significantly enhances the safety of LLMs while not degrading their usefulness. However, adding excessive safety samples may lead to LLMs rejecting safe prompts that superficially resemble unsafe ones. Fu et al. (2024b) employ a similar practice to improve the robustness of LLMs against malicious queries when processing long text. Another noteworthy work by Lyu et al. (2024) uncovers the crucial role of the prompt templates in preserving safety alignment. The proposed "Pure Tuning, Safe Testing" strategy aims to maintain the model's safety constraints while enhancing its performance by employing carefully designed prompts. This dual approach helps ensure that the fine-tuning process does not compromise the model's robustness and safety features.

Furthermore, to address the threat of extraction attacks, Ishibashi & Shimodaira (2024) propose a knowledge sanitization method that fine-tunes LLMs to generate innocuous responses such as "I don't know" when encountering sensitive data. This approach not only safeguards against the extraction of sensitive information but also maintains the overall performance of LLMs in various tasks. Additionally, Rosati et al. (2024) offer a defense mechanism that effectively prevents LLMs from being maliciously fine-tuned for harmful purposes. This mechanism works by removing information about harmful representations such that it is difficult to recover them during fine-tuning. More importantly, it is able to generalize across different subsets of harm that have not been seen during the defense process, substantially enhancing the robustness of LLMs to harmful fine-tuning attacks.

Pain Point: There is a well-known trade-off in enhancing LLM's instruction-following capabilities while ensuring they remain safe and reliable. On the other hand, fine-tuning itself can be used both to enhance the safety of LLMs by adding safety-related examples, and to attack LLMs by introducing adversarial examples. The mechanism proposed by Rosati et al. (2024), while seemingly promising, is limited to defending against supervised fine-tuning attacks in LLMs. Also, it requires paired safe and unsafe examples, which makes data collection more expensive and complex.

Future Work: Future work on safety fine-tuning perhaps should strive to achieve a win-win situation for both the safety and utility of LLMs. At the same time, future research should invest in stronger attack settings to emulate worst-case attacks during the fine-tuning process and investigate different types of harm, based on which more effective and comprehensive defense mechanisms need to be developed.

4.4 Post-Safety Alignment

Beyond the regular safety alignment process, post-safety alignment has emerged as a secondary safeguard for LLMs against various potential vulnerabilities. This ongoing process is concerned with seeking solutions from both outside and inside the backbone of LLMs to keep them from generating undesirable behaviors (Zhao et al., 2024c). Accordingly, relevant defense techniques have been developed to mitigate a majority of the safety risks posed by data privacy attacks, prompt injection attacks, and jailbreak attacks.

A significant branch of post-safety alignment techniques is machine unlearning (Si et al., 2023; Zhang et al., 2024a; Liu et al., 2024a). This paradigm involves selectively forgetting or erasing undesirable knowledge in LLMs, *e.g.,* copyrighted and user privacy content, expecting to mitigate risks associated with model outputs that are potentially influenced by biased or sensitive information. Under this principle, Yao et al. (2023b) have pioneered a work that uses only negative examples to unlearn LLMs, which is done by applying gradient ascent on the loss function of those undesirable samples. Subsequent studies (Lu et al., 2024; Zhang et al.,

2024b) have improved this approach by emphasizing the need to retain general knowledge while unlearning harmful knowledge. Instead of tuning all model parameters for unlearning, Chen & Yang (2023) introduce an efficient framework to eliminate the effect of unwanted data by designing separate lightweight unlearning layers. These layers learn to forget different sets of data under the guidance of a selective teacher-student objective. Moreover, Liu et al. (2024f) present a two-stage unlearning framework based on the concept of first isolating and then removing harmful knowledge in model parameters. This framework has been shown to effectively balance the trade-off between removing undesirable information and preserving utility.

Another line of research seeks to prevent LLMs from generating harmful content or resisting jailbreak attacks via steering the decoding process of LLMs. Prompt engineering (White et al., 2023; Chen et al., 2023; Phute et al., 2023; Suo, 2024), as a way to indirectly control the decoding of LLMs, has been preferred due to its ease of operation. To illustrate, Xie et al. (2023) present a simple yet effective technique to defend against various jailbreak attacks by encapsulating the user's query in a system prompt that reminds ChatGPT to respond responsibly. Further, Wei et al. (2023b) explore a strategy called "In-Context Defense" that teaches the LLM to resist jailbreaking by imitating a few examples of refusing harmful queries. On the other hand, directly manipulating the probability of generated tokens provides a more precise manner to bootstrap the LLM's decoding process. For instance, Zhong et al. (2024) propose a straightforward method based on the principle of contrastive decoding, which aims to boost the probability of desired safe outputs by suppressing undesired outputs. Besides, Xu et al. (2024a) develop a safety-aware decoding strategy to protect LLMs from jailbreak attacks. This strategy identifies safety disclaimers and amplifies their token probabilities while attenuating the probabilities of token sequences aligned with jailbreak attacks' objectives.

Pain Point: Although unlearning-based methods succeed in enhancing the safety of the backbone model, they also significantly exacerbate the issue of over-safety. Additionally, utilizing general data for distillation not only incurs extra training costs but also has limited effectiveness in maintaining utility. On the other hand, decoding-based approaches do not address the core problem of harmful output from LLMs, since the harmful knowledge remains within the model. Additionally, they also introduce extra costs during inference. As a result, safety enhancement made by existing approaches typically comes at the cost of significantly increasing over-safety and compromising utility, with the drawbacks outweighing the benefits.

Future Work: As discussed above, future research might have to consider how to effectively achieve the three objectives simultaneously, *i.e.,* safety enhancement, over-safety mitigation, and utility preservation. A recent work by Zhao et al. (2024c) has made a beneficial exploration toward this direction, but it is limited to small-scale LLMs (no more than 13B) and focuses on general safety issues. Therefore, we urge that more efforts should be devoted to developing post-safety alignment techniques that can alleviate over-safety and maintain utility, as well as the comprehensive evaluation of these techniques in a wider range of domains.

4.5 Model Merge

The methodology of model merging originates from the observation that different fine-tuned models initialized from the same pre-trained backbone model may share a part of the optimization trajectory and diverge on only a fraction of the model parameters oriented to different learning tasks. The parameters of the finely tuned models adapted to different tasks can be hence merged via arithmetic averaging to reach better generalization over out-of-domain input and deliver multi-task learning at the same time. In fact, the idea of merging model parameters to mitigate conflicts between different tasks has been employed and proved to be effective in Federated Learning and continual learning methods. Following the spirit, Zhao et al. (2024b); Kadhe et al. (2024) adopt the model merging methods to achieve a balance between unlearning the unsafe response, yet avoiding over-defensiveness as much as possible. They first assume to have a collection of input questions and well-tagged harmful response texts. They use gradient ascent using the harmful data collection over the backbone model to first compute the model parameter update of the backbone model, dedicated to preventing harmful answers. Furthermore, Zhao et al. (2024b) choose to perform gradient descent over the unaligned model using the harmful input-answer pairs to compute the model parameter update to mitigate over-defense. The two patches of model updates are then integrated using the model merging technique to derive a safety-aligned model balancing the safety alignment and utility.

This line of model merging techniques (Yu et al., 2024; Yadav et al., 2023; Zhang et al., 2023), though operationally simple, still lack deep theoretical investigation regarding two perspectives. First of all, it is unclear how the adversarially updated model parameters with the unlearning objective and tagged harmful responses are associated with the embeddings with safe response. It is possible that adversarial training with a limited set of harmful response texts is prone to overfitting and makes the model still vulnerable to further new jailbreaking prompts. Second, it is difficult to control the over-defensiveness of the merged model parameters. Merging the model parameters does not provide an explicit explanation of how the overdefensive response may be prevented.

5 Discussion

The field of large language models (LLMs) is indeed dynamic and rapidly evolving, with continuous advancements in both attacks and defenses. As attackers uncover new vulnerabilities and develop sophisticated exploits, defenders are required to respond with equally innovative countermeasures. This ongoing interaction between attackers and defenders is crucial for developing safe and reliable LLM systems.

Measuring attack effectiveness on LLMs is often more straightforward than evaluating the robustness of defense. Nevertheless, the effectiveness of defenses must be scrutinized carefully to avoid a false sense of security. Guaranteeing the defense strength is necessary to ensure that the defenses are not only theoretically sound but also practically effective against real-world threats in LLM systems.

Theoretical analysis of LLMs has indeed struggled to keep pace with their rapid growth and deployment. The sheer complexity and scale of these models pose significant challenges for theoretical frameworks that aim to understand and predict their behavior. While theoretical analysis provides valuable insights, it often falls short in capturing the nuanced performance of LLMs in practical applications.

Recent research has made strides by focusing on scaled case studies and examining specific features and inner mechanisms of LLMs (Lee et al., 2024; Allen-Zhu & Li, 2024; Templeton, 2024). These studies enhance our understanding of model behavior and contribute to improving model interpretability. For instance, the work inspired by (Xu et al., 2024c) emphasizes the need to explore fundamental questions about LLMs, such as the possibility of completely eliminating non-robust features.

In summary, advancing our understanding of LLMs requires a concerted effort to address both theoretical and practical challenges. Researchers are encouraged to investigate fundamental questions and refine both attack and defense strategies to keep pace with the rapid advancements in LLM technology.

References

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.2, knowledge manipulation, 2024. URL https://arxiv.org/abs/2309.14402.

Maksym Andriushchenko and Nicolas Flammarion. Does refusal training in llms generalize to the past tense?, 2024. URL https://arxiv.org/abs/2407.11969.

Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned llms with simple adaptive attacks, 2024. URL https://arxiv.org/abs/2404.02151.

Cem Anil, Esin Durmus, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Nina Rimsky, Meg Tong, Jesse Mu, Daniel Ford, et al. Many-shot jailbreaking. Technical report, Anthropic, April 2024. URL https://example.com/many-shot-jailbreaking. Accessed: 2024-08-12.

Stuart Armstrong. Using gpt-eliezer against chatgpt jailbreaking, 2022.

Yang Bai, Ge Pei, Jindong Gu, Yong Yang, and Xingjun Ma. Special characters attack: Toward scalable training data extraction from large language models, 2024. URL https://arxiv.org/abs/2405.05990.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom

Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

Tim Baumgärtner, Yang Gao, Dana Alon, and Donald Metzler. Best-of-venom: Attacking rlhf by injecting poisoned preference data, 2024. URL https://arxiv.org/abs/2404.05530.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *arXiv preprint arXiv:2309.07875*, 2023.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. Emergent and predictable memorization in large language models, 2023. URL https://arxiv.org/abs/2304.11158.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. *Evasion Attacks against Machine Learning at Test Time*, pp. 387–402. Springer Berlin Heidelberg, 2013. ISBN 9783642387098. doi: 10.1007/978-3-642-40994-3_25. URL http://dx.doi.org/10.1007/978-3-642-40994-3_25.

Hezekiah J. Branch, Jonathan Rodriguez Cefalu, Jeremy McHugh, Leyla Hujer, Aditya Bahl, Daniel del Castillo Iglesias, Ron Heichman, and Ramesh Darwishi. Evaluating the susceptibility of pre-trained language models via handcrafted adversarial examples, 2022. URL https://arxiv.org/abs/2209.02128.

Matt Burgess. Hackingchatgpt. the hacking of chatgpt is just getting started., 2023. https://www.wired.com/story/chatgpt-jailbreak-generative-ai-hacking/.

Aleksander Buszydlik, Karol Dobiczek, Michał Teodor Okoń, Konrad Skublicki, Philip Lippmann, and Jie Yang. Red teaming for large language models at scale: Tackling hallucinations on mathematics tasks, 2023. URL https://arxiv.org/abs/2401.00290.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017. doi: 10.1109/SP.2017.49.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks, 2019. URL https://arxiv.org/abs/1802.08232.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models, 2021. URL https://arxiv.org/abs/2012.07805.

Stephen Casper, Lennart Schulze, Oam Patel, and Dylan Hadfield-Menell. Defending against unforeseen failure modes with latent adversarial training, 2024. URL https://arxiv.org/abs/2403.05030.

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735*, 2023.

Canyu Chen and Kai Shu. Can llm-generated misinformation be detected?, 2024. URL https://arxiv.org/abs/2309.13788.

Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*, 2023.

Kyunghyun Cho. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

Arijit Ghosh Chowdhury, Md Mofijul Islam, Vaibhav Kumar, Faysal Hossain Shezan, Vaibhav Kumar, Vinija Jain, and Aman Chadha. Breaking down the defenses: A comparative survey of attacks on large language models, 2024. URL https://arxiv.org/abs/2403.04786.

Jon Christian. Amazing "jailbreak" bypasses chatgpt's ethics safeguards., 2023. https://futurism.com/amazing-jailbreak-chatgpt.

Jing Cui, Yufei Han, Yuzhe Ma, Jianbin Jiao, and Junge Zhang. Badrl: Sparse targeted backdoor attack against reinforcement learning, 2023. URL https://arxiv.org/abs/2312.12585.

Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. Security and privacy challenges of large language models: A survey. *arXiv preprint arXiv:2402.00888*, 2024.

Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. Masterkey: Automated jailbreaking of large language model chatbots. In *Proceedings 2024 Network and Distributed System Security Symposium*, NDSS 2024. Internet Society, 2024. doi: 10.14722/ndss.2024.24188. URL http://dx.doi.org/10.14722/ndss.2024.24188.

Xiaoning Feng, Xiaohong Han, Simin Chen, and Wei Yang. Llmeffichecker: Understanding and testing efficiency degradation of large language models, 2024. URL https://arxiv.org/abs/2210.03696.

Colin Fraser. Master thread of ways i have discovered to get chatgpt to output text that it's not supposed to, including bigotry, urls and personal information, and more., 2023. https://twitter.com/colinfraser/status/1630763219450212355.

Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. Practical membership inference attacks against fine-tuned large language models via self-prompt calibration, 2024a. URL https://arxiv.org/abs/2311.06062.

Yu Fu, Wen Xiao, Jia Chen, Jiachen Li, Evangelos Papalexakis, Aichi Chien, and Yue Dong. Cross-task defense: Instruction-tuning llms for content safety. *arXiv preprint arXiv:2405.15202*, 2024b.

Pranav Gade, Simon Lermen, Charlie Rogers-Smith, and Jeffrey Ladish. Badllama: cheaply removing safety fine-tuning from llama 2-chat 13b, 2024. URL https://arxiv.org/abs/2311.00117.

Filippo Galli, Luca Melis, and Tommaso Cucinotta. Noisy neighbors: Efficient membership inference attacks against llms, 2024. URL https://arxiv.org/abs/2406.16565.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

José Gonçalves, Tiago Dias, Eva Maia, and Isabel Praça. Scope: Evaluating llms for software vulnerability detection. *arXiv preprint arXiv:2407.14372*, 2024.

Riley Goodside. Exploiting gpt-3 prompts with malicious inputs that order the model to ignore its previous directions, 2023.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection, 2023. URL https://arxiv.org/abs/2302.12173.

Kai Greshakeblog. Indirect prompt injection threats., 2023. URL https://greshake.github.io/.

Aman Gupta, Anup Shirgaonkar, Angels de Luis Balaguer, Bruno Silva, Daniel Holstein, Dawei Li, Jennifer Marsman, Leonardo O Nunes, Mahsa Rouzbahman, Morris Sharp, et al. Rag vs fine-tuning: Pipelines, tradeoffs, and a case study on agriculture. *arXiv preprint arXiv:2401.08406*, 2024.

Danny Halawi, Alexander Wei, Eric Wallace, Tony T. Wang, Nika Haghtalab, and Jacob Steinhardt. Covert malicious finetuning: Challenges in safeguarding llm adaptation, 2024. URL https://arxiv.org/abs/2406.20053.

Sanghyun Hong, Yiğitcan Kaya, Ionuţ-Vlad Modoranu, and Tudor Dumitraş. A panda? no, it's a sloth: Slowdown attacks on adaptive multi-exit neural network inference, 2021. URL https://arxiv.org/abs/2010.02432.

Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. Catastrophic jailbreak of open-source llms via exploiting generation, 2023. URL https://arxiv.org/abs/2310.06987.

Injection Guide. Adversarial prompting guide, 2023. URL https://www.promptingguide.ai/risks/adversarial.

Yoichi Ishibashi and Hidetoshi Shimodaira. Knowledge sanitization of large language models, 2024. URL https://arxiv.org/abs/2309.11852.

Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P. Dick, Hidenori Tanaka, Edward Grefenstette, Tim Rocktäschel, and David Scott Krueger. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks, 2023. URL https://arxiv.org/abs/2311.12786.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding, 2020. URL https://arxiv.org/abs/1909.10351.

Jeevesh Juneja, Rachit Bansal, Kyunghyun Cho, João Sedoc, and Naomi Saphra. Linear connectivity reveals generalization strategies, 2023. URL https://arxiv.org/abs/2205.12411.

Swanand Ravindra Kadhe, Farhan Ahmed, Dennis Wei, Nathalie Baracaldo, and Inkit Padhi. Split, unlearn, merge: Leveraging data attributes for more effective unlearning in llms, 2024. URL https://arxiv.org/abs/2406.11780.

Divyanshu Kumar, Anurakt Kumar, Sahil Agarwal, and Prashanth Harshangi. Increased llm vulnerabilities from fine-tuning and quantization, 2024a. URL https://arxiv.org/abs/2404.04392.

Surender Suresh Kumar, ML Cummings, and Alexander Stimpson. Strengthening llm trust boundaries: A survey of prompt injection attacks surender suresh kumar dr. ml cummings dr. alexander stimpson. In *2024 IEEE 4th International Conference on Human-Machine Systems (ICHMS)*, pp. 1–6. IEEE, 2024b.

Raz Lapid, Ron Langberg, and Moshe Sipper. Open sesame! universal black box jailbreaking of large language models, 2024. URL https://arxiv.org/abs/2309.01446.

Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin Wattenberg, Jonathan K. Kummerfeld, and Rada Mihalcea. A mechanistic understanding of alignment algorithms: A case study on dpo and toxicity, 2024. URL https://arxiv.org/abs/2401.01967.

Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, Jie Huang, Fanpu Meng, and Yangqiu Song. Multi-step jailbreaking privacy attacks on chatgpt, 2023. URL https://arxiv.org/abs/2304.05197.

Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. Fq-vit: Post-training quantization for fully quantized vision transformer. *arXiv preprint arXiv:2111.13824*, 2021.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. Rethinking machine unlearning for large language models, 2024a. URL https://arxiv.org/abs/2402.08787.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. Adversarial training for large neural language models, 2020. URL https://arxiv.org/abs/2004.08994.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models, 2024b. URL https://arxiv.org/abs/2310.04451.

Xiaogeng Liu, Zhiyuan Yu, Yizhe Zhang, Ning Zhang, and Chaowei Xiao. Automatic and universal prompt injection attacks against large language models, 2024c. URL https://arxiv.org/abs/2403.04957.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. Prompt injection attack against llm-integrated applications, 2024d. URL https://arxiv.org/abs/2306.05499.

Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. Formalizing and benchmarking prompt injection attacks and defenses, 2024e. URL https://arxiv.org/abs/2310.12815.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. Towards safer large language models through machine unlearning. *arXiv preprint arXiv:2402.10058*, 2024f.

Weikai Lu, Ziqian Zeng, Jianwei Wang, Zhengdong Lu, Zelin Chen, Huiping Zhuang, and Cen Chen. Eraser: Jailbreaking defense in large language models via unlearning harmful knowledge. *arXiv preprint arXiv:2404.05880*, 2024.

Ekdeep Singh Lubana, Eric J. Bigelow, Robert P. Dick, David Krueger, and Hidenori Tanaka. Mechanistic mode connectivity, 2023. URL https://arxiv.org/abs/2211.08422.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2024. URL https://arxiv.org/abs/2308.08747.

Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. Keeping llms aligned after fine-tuning: The crucial role of prompt templates, 2024. URL https://arxiv.org/abs/2402.18540.

Andrew A Mahyari. Harnessing the power of llms in source code vulnerability detection. *arXiv preprint arXiv:2408.03489*, 2024.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison, 2023. URL https://arxiv.org/abs/2305.18462.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically, 2024. URL https://arxiv.org/abs/2312.02119.

Yifei Min, Lin Chen, and Amin Karbasi. The curious case of adversarially robust models: More data can help, double descend, or hurt generalization, 2020. URL https://arxiv.org/abs/2002.11080.

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks, 2022. URL https://arxiv.org/abs/2203.03929.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale, 2022. URL https://arxiv.org/abs/2206.06520.

Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning, 2018. URL https://arxiv.org/abs/1704.03976.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models, 2023. URL https://arxiv.org/abs/2311.17035.

Myung Gyo Oh, Leo Hyun Park, Jaeuk Kim, Jaewoo Park, and Taekyoung Kwon. Membership inference attacks with token-level deduplication on korean language models. *IEEE Access*, 11:10207–10217, 2023. doi: 10.1109/ACCESS.2023.3239668.

OpenAI. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings, 2015. URL https://arxiv.org/abs/1511.07528.

Pankayaraj Pathmanathan, Souradip Chakraborty, Xiangyu Liu, Yongyuan Liang, and Furong Huang. Is poisoning a real threat to llm alignment? maybe more so than you think, 2024. URL https://arxiv.org/abs/2406.12091.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4, 2023. URL https://arxiv.org/abs/2304.03277.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models, 2022. URL https://arxiv.org/abs/2202.03286.

Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models, 2022. URL https://arxiv.org/abs/2211.09527.

Mansi Phute, Alec Helbling, Matthew Daniel Hull, ShengYun Peng, Sebastian Szyller, Cory Cornelius, and Duen Horng Chau. Llm self defense: By self examination, llms know they are being tricked. In *The Second Tiny Papers Track at ICLR 2024*, 2023.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking, 2024a. URL https://arxiv.org/abs/2402.14811.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking, 2024b. URL https://arxiv.org/abs/2402.14811.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to!, 2023a. URL https://arxiv.org/abs/2310.03693.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to!, 2023b. URL https://arxiv.org/abs/2310.03693.

Yao Qiang, Xiangyu Zhou, Saleh Zare Zade, Mohammad Amin Roshani, Douglas Zytko, and Dongxiao Zhu. Learning to poison large language models during instruction tuning, 2024. URL https://arxiv.org/abs/2402.13459.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy, 2020. URL https://arxiv.org/abs/2002.10716.

Javier Rando and Florian Tramèr. Universal jailbreak backdoors from poisoned human feedback, 2024. URL https://arxiv.org/abs/2311.14455.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist, 2020. URL https://arxiv.org/abs/2005.04118.

Domenic Rosati, Jan Wehner, Kai Williams, Łukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. Representation noising effectively prevents harmful fine-tuning on llms. *arXiv preprint arXiv:2405.14577*, 2024.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. Hatecheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.4. URL http://dx.doi.org/10.18653/v1/2021.acl-long.4.

Roman Samoilenko. New prompt injection attack on chatgpt web version. markdown images can steal your chat data., 2023a.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL https://arxiv.org/abs/1910.01108.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023. URL https://arxiv.org/abs/2302.04761.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Staff Seclify. Prompt injection cheat sheet: How to manipulate ai language models., 2023. URL https://blog.seclify.com/prompt-injection-cheat-sheet/.

Erfan Shayegani, Yue Dong, and Nael Abu-Ghazaleh. Plug and pray: Exploiting off-the-shelf components of multi-modal models. *arXiv preprint arXiv:2307.14539*, 2023a.

Erfan Shayegani, Md Abdullah Al Mamun, Yu Fu, Pedram Zaree, Yue Dong, and Nael Abu-Ghazaleh. Survey of vulnerabilities in large language models revealed by adversarial attacks, 2023b. URL https://arxiv.org/abs/2310.10844.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face, 2023. URL https://arxiv.org/abs/2303.17580.

Jiawen Shi, Yixin Liu, Pan Zhou, and Lichao Sun. Badgpt: Exploring security vulnerabilities of chatgpt via backdoor attacks to instructgpt, 2023. URL https://arxiv.org/abs/2304.12298.

Manli Shu, Jiongxiao Wang, Chen Zhu, Jonas Geiping, Chaowei Xiao, and Tom Goldstein. On the exploitability of instruction tuning, 2023. URL https://arxiv.org/abs/2306.17194.

Nianwen Si, Hao Zhang, Heyu Chang, Wenlin Zhang, Dan Qu, and Weiqiang Zhang. Knowledge unlearning for llms: Tasks, methods, and challenges, 2023. URL https://arxiv.org/abs/2311.15766.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking, 2022. URL https://arxiv.org/abs/2209.13085.

Robin Staab, Mark Vero, Mislav Balunović, and Martin Vechev. Beyond memorization: Violating privacy via inference with large language models, 2024. URL https://arxiv.org/abs/2310.07298.

Xuchen Suo. Signed-prompt: A new approach to prevent prompt injection attacks against llm-integrated applications. *arXiv preprint arXiv:2401.07612*, 2024.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014a. URL https://arxiv.org/abs/1312.6199.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014b. URL https://arxiv.org/abs/1312.6199.

Adly Templeton. *Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet*. Anthropic, 2024.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. Poisoning language models during instruction tuning, 2023. URL https://arxiv.org/abs/2305.00944.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey, 2023. URL https://arxiv.org/abs/2310.16218.

Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the importance of difficulty calibration in membership inference attacks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=3eIrli0TwQ.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail?, 2023a. URL https://arxiv.org/abs/2307.02483.

Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. Assessing the brittleness of safety alignment via pruning and low-rank modifications, 2024. URL https://arxiv.org/abs/2402.05162.

Zeming Wei, Yifei Wang, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023b.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*, 2023.

Simon Willison. Prompt injection series., 2022b. URL https://simonwillison.net/series/prompt-injection/.

Red Wunderwuzzi. Ai injections: Direct and indirect prompt injections and their implications., 2023.

Sophie Xhonneux, Alessandro Sordoni, Stephan Günnemann, Gauthier Gidel, and Leo Schwinn. Efficient adversarial training in llms with continuous attacks, 2024. URL https://arxiv.org/abs/2405.15589.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pp. 38087–38099. PMLR, 2023.

Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5(12): 1486–1496, 2023.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. Bot-adversarial dialogue for safe conversational agents. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2950–2968, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.235. URL https://aclanthology.org/2021.naacl-main.235.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. *arXiv preprint arXiv:2402.08983*, 2024a.

Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li, and Stjepan Picek. Llm jailbreak attack versus defense techniques–a comprehensive study. *arXiv preprint arXiv:2402.13457*, 2024b.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models, 2024c. URL https://arxiv.org/abs/2401.11817.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1.

Biwei Yan, Kun Li, Minghui Xu, Yueyan Dong, Yue Zhang, Zhaochun Ren, and Xiuzhen Cheng. On protecting the data privacy of large language models (llms): A survey, 2024a. URL https://arxiv.org/abs/2403.05156.

Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. Backdooring instruction-tuned large language models with virtual prompt injection, 2024b. URL https://arxiv.org/abs/2307.16888.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*, 2024a.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training, 2024b. URL https://arxiv.org/abs/2312.06635.

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models, 2023a. URL https://arxiv.org/abs/2310.02949.

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. Shadow alignment: The ease of subverting safely-aligned language models, 2023b. URL https://arxiv.org/abs/2310.02949.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023a. URL https://arxiv.org/abs/2305.10601.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 4(2): 100211, 2024. ISSN 2667-2952. doi: https://doi.org/10.1016/j.hcc.2024.100211. URL https://www.sciencedirect.com/science/article/pii/S266729522400014X.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *arXiv preprint arXiv:2310.10683*, 2023b.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, pp. 3093–3106, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450394505. doi: 10.1145/3548606.3560675. URL https://doi.org/10.1145/3548606.3560675.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *The 41st International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2311.03099.

Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher, 2024. URL https://arxiv.org/abs/2308.06463.

Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. Removing rlhf protections in gpt-4 via fine-tuning, 2024. URL https://arxiv.org/abs/2311.05553.

Dawen Zhang, Pamela Finckenberg-Broman, Thong Hoang, Shidong Pan, Zhenchang Xing, Mark Staples, and Xiwei Xu. Right to be forgotten in the era of large language models: Implications, challenges, and solutions, 2024a. URL https://arxiv.org/abs/2307.03941.

Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. Composing parameter-efficient modules with arithmetic operation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=5r3e27I9Gy.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024b.

Jiachen Zhao, Zhun Deng, David Madras, James Zou, and Mengye Ren. Learning and forgetting unsafe examples in large language models, 2024a. URL https://arxiv.org/abs/2312.12736.

Weixiang Zhao, Yulin Hu, Zhuojun Li, Yang Deng, Yanyan Zhao, Bing Qin, and Tat-Seng Chua. Towards comprehensive and efficient post safety alignment of large language models via safety patching, 2024b. URL https://arxiv.org/abs/2405.13820.

Weixiang Zhao, Yulin Hu, Zhuojun Li, Yang Deng, Yanyan Zhao, Bing Qin, and Tat-Seng Chua. Towards comprehensive and efficient post safety alignment of large language models via safety patching. *arXiv preprint arXiv:2405.13820*, 2024c.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. Rose doesn't do that: Boosting the safety of instruction-tuned large language models with reverse prompt contrastive decoding. *arXiv preprint arXiv:2402.11889*, 2024.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. URL https://arxiv.org/abs/2307.15043.