Building Safe GenAI Applications: An End-to-End Overview of Red Teaming for Large Language Models

Alberto Purpura*, **Sahil Wadhwa***, **Jesse Zymet***, **Akshay Gupta, Andy Luo,**
Melissa Kazemi Rad, Swapnil Shinde, Mohammad Shahed Sorower
Capital One, AI Foundations
{alberto.purpura, sahil.wadhwa, jesse.zymet, akshay.gupta3, andy.luo,
 melissa.kazemirad, swapnil.shinde2, mohammad.sorower}@capitalone.com

Abstract

The rapid growth of Large Language Models (LLMs) presents significant privacy, security, and ethical concerns. While much research has proposed methods for defending LLM systems against misuse by malicious actors, researchers have recently complemented these efforts with an offensive approach that involves *red teaming*, i.e., proactively attacking LLMs with the purpose of identifying their vulnerabilities. This paper provides a concise and practical overview of the LLM red teaming literature, structured so as to describe a multi-component system end-to-end. To motivate red teaming we survey the initial safety needs of some high-profile LLMs, and then dive into the different components of a red teaming system as well as software packages for implementing them. We cover various attack methods, strategies for attack-success evaluation, metrics for assessing experiment outcomes, as well as a host of other considerations. Our survey will be useful for any reader who wants to rapidly obtain a grasp of the major red teaming concepts for their own use in practical applications.

1 Introduction

The popularity and widespread adoption of Large Language Models (LLMs) has been transformative across many industries, ushering in new possibilities for enhancing productivity, decision-making, and user engagement. LLMs are contributing significantly to fields such as finance, healthcare, and legal services where they are being leveraged for tasks such as customer servicing support, clinical notes and contract analysis. However, the increasing reliance on LLMs brings with it a critical and challenging ethical-moral responsibility: ensuring that the deployed system responds to any possible input in safe or otherwise desirable ways. While LLMs offer remarkable capabilities, they are also

vulnerable to various forms of misuse. Such attacks could provoke LLMs to generate misinformative, biased, or toxic content (Abid et al., 2021; Lin et al., 2022) or expose private information (Carlini et al., 2021). Microsoft's Tay, in a high-profile case, was successfully provoked by attackers to send racist or sexually-charged tweets to a large audience (Lee, 2016). A great deal of research on improving LLM safety has been conducted from a defensive standpoint, with investigators developing methods for guardrailing LLMs against potential attacks (Dong et al., 2024). These attacks, however, must be identified beforehand, which has proven to be challenging – e.g., GPT-4 was vulnerable to attacks absent from its safety training that were written in low-resource languages (Yong et al., 2024). Investigators have hence turned to complementing defensive efforts with an offensive approach to LLM safety, proposing strategies for *red teaming* LLMs, i.e., proactively attacking or testing LLMs with the purpose of identifying their vulnerabilities. Red teaming is useful for any organization that aims not only to productionize some LLM-supported system, but to effectively anticipate threats to their system and safeguard against them before production.

While prior reviews of LLM red teaming focused on serving as an encyclopedic taxonomic resource, e.g., of attack methodologies (Lin et al., 2024), we anticipate a wide need for a concise and practical overview geared toward readers who want to rapidly grasp the major concepts and components of a red teaming system and available software tools that have emerged, for example to devise and implement a system of their own. The purpose of this paper is to provide such an overview: one that balances comprehensive treatment of research with conciseness, and structures the exposition to describe a multi-component system end-to-end. Figure 1 provides an illustration of the framework and its components, the latter of which are covered in

---
*Equal contribution

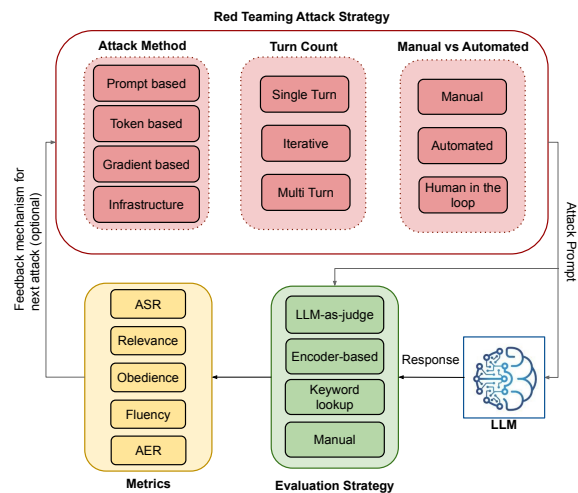Sections 4, 5 and 6. After covering related work,

Figure 1: Core components of a red teaming system.

we survey a few case studies from the tech industry, highlighting problems that motivate the field of red teaming. We then dive into the central components that make up a complete red teaming system, reviewing popular methods, software packages, and other resources that have emerged to support these components. We cover various attack strategies, with attention paid to categorizing particular methods and distinguishing single-turn from multi-turn attacking, manual from automated attacking, and different varieties of automated attacking. We then dive into popular approaches to attack success evaluation, as well as safety metrics for assessing overall experimental outcomes. We discuss a number of publicly available resources for red teaming, including software packages and datasets. We also touch briefly upon guardrailing steps commonly taken after red teaming. Finally, we close with future directions that we judge to constitute some of the most impactful opportunities for progress, including strategies for adapting automated attackers to generate more relevant and diverse sets of attacks, including multi-turn ones.

2 Related work

Recent literature has explored various facets of LLM red teaming, offering valuable insights into the rapidly evolving field. Some organizations aim to provide up-to-date informational materials geared toward helping developers and web-security practitioners secure their particular applications (MITRE, 2024; Commons, 2024). For example, the OWASP Foundation published the OWASP Top 10 (OWASP Foundation, 2025), a document

that describes, as deemed by common consensus, some of the most major threats to the security of LLM-supported applications, and provides mitigation strategies. On the academic side, Feffer et al. (2024) provide a high-level overview of and stance on red teaming practices, indexing on particular aspects of the literature to argue that the red teaming community lacks consensus around scope, structure, and evaluation of red-teaming. Verma et al. (2024) operationalize a threat model for red teaming, providing a taxonomy based on entry points in the LLM lifecycle. Rawat et al. (2024) provide a practitioner's viewpoint of challenges within LLM red teaming and emphasizes the context-dependent nature of vulnerabilities, and introduces a taxonomy of single-turn, prompt-based attacks. Mo et al. (2024) develop a taxonomy of attacks against language agents in particular – i.e., systems equipped with additional capacities for reasoning, planning, and task completion. Shi et al. (2024) offer a comprehensive survey of LLM safety more broadly, encompassing various risks beyond attacks, including value misalignment and autonomous AI risks. But perhaps the most extensive treatment to date specifically of LLM red teaming is given in Lin et al. (2024), which provides a fine-grained taxonomy of attack strategies grounded in LLM capabilities as well as several mitigating strategies, an overview of attack success evaluation strategies, and a framework that unifies attack-search strategies for automated red teaming. While insightful, the latter two articles' extensive lengths would be prohibitive for readers who seek a more concise overview of major red teaming concepts and trends. We see these papers as valuable in their own right, but anticipate the need for a resource that balances broad representation of the literature with concise exposition.

3 Policies on LLM Safety

Policies and risk mitigation strategies devised for ensuring the proper use of LLM-driven products have been crucial to their safety and success. This section serves to motivate LLM red teaming and LLM safety, providing a brief survey of some major safety considerations and policies from different leading LLM providers within industry along with risk mitigation strategies they have taken. These policies play a key role in shaping the goals for an adequate LLM safety solution, of which red teaming constitutes a critical part. While govern-

ments have taken steps to address LLM risks,[1] various organizations have established their own safety guidelines, leading to diverse priorities and approaches. As previously mentioned, organizations such as MITRE, MLCommons, and OWASP (MITRE, 2024; Commons, 2024; OWASP Foundation, 2025; Vidgen et al., 2024) have published materials to help practitioners to secure their LLM applications; these materials form a helpful basis for policy formulation, as they categorize risks by severity and provide recommendations for evaluating AI safety . OpenAI was an early pioneer of LLM use policies, emphasizing legal compliance and protection of privacy (OpenAI, 2023b, 2024, 2023c,d), and, along these ends, has employed both red teaming and guardrailing to prevent users from soliciting various kinds of harmful responses from their models (OpenAI, 2023e). Meta, as a large-scale social media platform, addresses risks such as election interference in their use policies (Meta, 2024a; Meta, 2024b, 2023). Anthropic's policies emphasize ethical alignment in particular (Anthropic, 2023a, 2022; Anthropic, 2024, 2023b), and they employ guardrailing and red teaming practices and fairness evaluations to develop models with unbiased decision-making capacities.

4 Categorizing attacks against LLMs

In this section, we categorize and describe various strategies for attacking LLMs. Our analysis reflects the reality that the LLM's attack surface is high context-dependent and influenced by many factors including target-system type, its infrastructure, conversational history, and access privileges.

4.1 Attack Methods

Here, we categorize and describe various methods that users have employed to attack LLMs. We include a more extensive survey in the Appendix.

Prompt-based attacks exploit LLMs by crafting malicious prompts to circumvent the model's safeguards. They are especially common in *closed-box* systems, such as OpenAI's ChatGPT (OpenAI, 2023a) and Google's Gemini (Team et al., 2024), where attackers interact solely with the external interface of the model, lacking access to its internal

weights or system-level configurations. Techniques include prompt injection (Liu et al., 2023; Mehrotra et al., 2023), which disguises malicious instructions as benign inputs, and jailbreaking (Wei et al., 2024; Chao et al., 2024a), which provoke the target LLM to ignore its safeguards. Recently, these major categories have been subdivided into a growing set of more granular categories such as indirect prompt injection (Greshake et al., 2023), refusal suppression and style injection (Zhou et al., 2024a; Geiping et al., 2024; Guo et al., 2024), prompt-level obfuscation (Pape et al., 2024), and many-shot jailbreaking (Anil et al., 2024). Some of these attacks utilize personification techniques such as role-playing to influence the target LLM into adopting a specific persona (Zhang et al., 2024b; Shah et al., 2023). This manipulation can lead the LLM to relax its ethical constraints and safeguards – e.g., Chao et al. (2024b) highlight the relative effectiveness of role-playing for jailbreaking LLMs in their PAIR paradigm. Similarly, Shen et al. (2024) introduce a notable role-playing character, DAN (Do Anything Now), which exploits the LLM's internal permissions, granting elevated privileges (e.g., Admin Privileges) to bypass safety mechanisms.

Token-based attacks are designed to generate variants of existing malicious prompts in order to identify novel successful attacks. Early approaches replace characters, tokens, or entire words within prompts with synonyms or symbols with comparable usage (Rocamora et al.; Morris et al., 2020b); others simply affix symbolic material to prompts, which can confuse the system and cause it to let its guard down (Wallace et al., 2021). More recent approaches change the text encoding (Bai et al., 2024), translate it into low-resource languages (Wang et al., 2024a; Deng et al., 2024b; Yong et al., 2024) or use ciphers (Inie et al., 2024; Yuan et al., 2024). By design, these strategies are not always interpretable, making it challenging to analyze how or why a specific sequence successfully bypasses the model's safeguards.

Gradient-based attacks are designed instead for when attackers have access to a model's parameters – as in an *open-box* system – such as its weights, activations, and hyperparameters. Such attacks apply gradient descent to find the most effective attack prompts (Shin et al., 2020; Geisler et al., 2024; Wichers et al., 2024). A few gradient-based approaches have also shown promising generalization power when applied to closed-box systems (Zou et al., 2023). These attacks are entirely unin-

---

terpretable and lack any semantic meaning (Morris et al., 2020a), and are commonly blocked using perplexity-based solutions (Jain et al., 2023).

Infrastructure attacks involve injecting material into, extracting material from, or somehow modifying the structures that support the target LLM. One subset of such attacks includes *data poisoning attacks* (or *backdoor attacks*), which involve injecting problematic data or documents into the ecosystem (Yao et al., 2024b,a). For example, the attacker might add malicious documents to an external knowledge source or API that an LLM is querying to formulate responses at runtime. The problem is often discussed in the context of agents and Retrieval Augmented Generation (RAG) pipelines — LLMs that are integrated with and call upon knowledge bases, APIs, and other software tools in order to execute tasks — since they are often susceptible to indirect prompt injection (Greshake et al., 2023), in which the malicious signal is injected into system-supporting knowledge bases or other infrastructure that then manifest an attack at retrieval time. For example, attackers could inject malicious material into external knowledge bases (e.g., Wikipedia or Wikidata) that the target LLM would then call upon to address questions. Alternatively, attackers could inject data into the model's training set, provided it is available, leading to problematic post-training behaviors. *Data extraction attacks* and *model extraction attacks*, on the other hand, involve extracting model data or aspects of the model itself. Data extraction attacks take place when internal data that supports the model, which may contain private or sensitive information, is unlawfully extracted (Carlini et al., 2021). Beyond data, LLMs could fall prey to model theft attacks in which the model parameters themselves are extracted for unauthorized copying or use, violating intellectual property rights (Kariyappa et al., 2021; Yao et al., 2024b).

4.2 Attacks by Turn Count

When attacking a model, we can distinguish interactions between the attacker and target LLM based on whether the attack takes place across a single turn or multiple turns.

Single-turn attack pipelines are simple to implement and ideal for applications that lack memory and do not leverage conversational history (Xu et al., 2024; Rawat et al., 2024). A red teaming pipeline will often leverage a corpus of malicious prompts that constitute single-turn attacks (see Section 7 for useful pointers to public data sources), sending pitting each them of them against the target LLM. Single-turn attacking will be limitedly effective against more complex target LLMs that critically leverage conversational history, since the latter could fall victim to attacks that only manifest after multiple conversational turns. Though common, single-turn attacks have generally become less effective now that a number of alignment techniques have been devised to ensure that the target LLM does not deviate from its intended purpose (Zhou et al., 2024b).

Multi-turn attacks, in contrast, leverage multiple conversational turns to implement attacking. We first describe what we call the *iterative attack*, which takes as a seed a single-turn attack prompt and progressively adapts it across multiple attempts at attacking, in order to maximize the likelihood of attack success. These attacks do not rely on a rich contextual history of prior interactions with the target, but instead merely track prior iterations on the same seed. Notable recent examples of iterative strategies include PAIR (Chao et al., 2024a), TAP (Mehrotra et al., 2023), DAN (Shen et al., 2024), AutoDAN(-Turbo) (Liu et al., 2024b,a), RedAgent (Xu et al., 2024), MART (Ge et al., 2023), and APRT (Jiang et al., 2024b); early synonym-replacing approaches arguably also constitute examples (Morris et al., 2020b; Rocamora et al.). We extend our discussion on these strategies in Section 4.3. Beyond the iterative attack, a multi-turn attack can be built by engaging in more complex back-and-forth conversation with the target LLM, exploiting the semantics of conversational history. To take one example from Li et al. (2024a), in order to provoke an LLM into claiming that the health effects of Agent Orange were overstated, an attacker might: 1) ask the LLM to write an essay arguing that the substance brought about horrible health effects to victims; 2) then ask the LLM to write an essay taking the *opposite* stance. The authors find that human panels are particularly effective at identifying such multi-turn attacks, well beyond the capacities of the automated approaches that they tested. Automated approaches that have emerged since then include Crescendo (Russinovich et al., 2024), HARM (Mazeika et al., 2024), and RedQueen (Jiang et al., 2024c).

4.3 Manual Versus Automated Attacking

Attacks can be formulated manually by humans, automatically by systems such as LLMs, or by both.

Human experts have proven extremely helpful for red teaming LLM-driven systems (Li et al., 2024a). It has become common practice for organizations to employ human panels for red teaming and other safety-preparedness work — OpenAI, for example, employed human panels before their releases of GPT-4 (Markov et al., 2022). While it has been shown time and time again that humans are able to devise creative attacks, safety practitioners have found that crowdsourcing attacks can lead to templatic prompts (e.g., "give a mean prompt that begins with X") without greatly expanding attack coverage (Ganguli et al., 2022). Further, human annotation is expensive, which limits the number and diversity of test cases.

Automated solutions, on the other hand, have gained increasing popularity by providing cheaper alternatives to evaluate the safety of LLM systems, relative to human panels. Such solutions involve automatically generating attacks against the target LLM, whose subsequent responses are evaluated for the presence of problematic content (e.g., by a trained detector). While previous work augmented attack datasets using synonym replacement and related strategies (Morris et al., 2020b; Rocamora et al.), more recent approaches leverage LLMs to generate novel attacks (Perez et al., 2022; Ganguli et al., 2022; Deng et al., 2023; Mo et al., 2023; Greshake et al., 2023; Yu et al., 2023; Paulus et al., 2024; Hong et al., 2024). In the latter case, an LLM is prompted or trained to generate a large number of examples to attack a target LLM. Since their inception, LLM-driven attack generators have been employed in whole ecosystems for automated, iterative attacking, as in PAIR (Chao et al., 2024a), TAP (Mehrotra et al., 2023), DAN (Shen et al., 2024), AutoDAN(-Turbo) (Liu et al., 2024b,a), and RedAgent (Xu et al., 2024). These solutions are unified by a common framework: an LLM-driven attacker generates an initial attack that is submitted to the target LLM; an LLM-driven evaluator then evaluates the interaction; the evaluator's signal is then passed back to the attack generator, which adapts the initial attack in some way in an attempt to increase attack success likelihood. Attack generation, response evaluation, and adaptation repeat in an iterating loop across multiple sequenced rounds. Here we single out RedAgent (Xu et al., 2024), which additionally formulates attacks against agents that are specific to the latter's infrastructural context. Other more complex ecosystems such as MART (Ge et al., 2023) and APRT

(Jiang et al., 2024b) were developed based on the aforementioned iterative framework but set up an adversarial environment, in which the target LLM jointly adapts its defense strategies together with the attack generator, so that the target LLM — now possessing strengthened defenses — can be used for downstream applications. Finally, frameworks like Crescendo (Russinovich et al., 2024), HARM (Zhang et al., 2024a), and RedQueen (Jiang et al., 2024c) support automated generation of more complex multi-turn attacks that exploit the semantics of the conversational history. Crescendo, for example, escalates attacks based on benign questions from prior turns — e.g., soliciting the recipe for a Molotov cocktail by first asking about its history and then about how it was historically made.

Human-in-the-loop solutions can involve humans guiding automated attack generation. Radharapu et al. (2023), for example, propose AART (AI-Assisted Red Teaming), a framework that employs automated attack generation in which humans help to select relevant attacks or filtering out those that are not likely to be successful. In addition to systems in which humans fundamentally aid AI generators, a number of AI-supported safety suites have been developed to assist *humans* to efficiently conduct red teaming and identify vulnerabilities (Wallace et al., 2019; Ziegler et al., 2022).

5 Evaluating Attack Success

The red teaming literature supplies various approaches to assessing based the target LLM's response whether an attack was successful.

Keyword-based (or lexical) evaluation methods attempt to match an LLM's response against a list of words, phrases, or other kind of regular expression (Derczynski et al., 2024). This approach is easily controllable and practitioners can expand or contract keyword lists as they see fit. On the other hand, this solution lacks insight into the general semantics of the response, and does not generalize to concepts that are not expressed in the keyword list (Moser et al., 2007).

Encoder-based text classifiers provide a more robust and specializable alternative to keyword-based approaches. For example, many practitioners have trained some variety of BERT classifier (Devlin et al., 2019; Liu et al., 2019; Caselli et al., 2021) to detect harmful responses (e.g., Yu et al. (2023); Derczynski et al. (2024)). However, these models often require training on domain-specific

data or a certain kind of harm to improve performance (Perez et al., 2022), and struggle to generalize to new harms without diverse training sets (Askell et al., 2021). In contrast with LLMs-as-judges, this limits their applicability to scenarios where data are available and efficiency and cost are less of a concern.

LLMs-as-Judges, on the other hand, are often leveraged due to their low barrier of entry and impressive performance (Zheng et al., 2023). Such an approach would prompt an LLM, separate from the attack generator, to judge target system responses or even attack-response pairs (e.g., Munoz et al. (2024)). Prior judges have returned binary assessments, scores on a 5-point scale, or continuous values (Shah et al., 2023; Zheng et al., 2023; Jones et al., 2024; Wang et al., 2023b). Prompting the LLM to respond with only a quantitative judgment has been shown to limit reasoning (Hao et al., 2024), and so they are often instructed to provide additional rationale (Sun et al., 2023; Wang et al., 2023c). Generic LLMs can perform poorly at providing domain-specific judgments (e.g., those about a financial context) (Dubey et al., 2024; Jiang et al., 2024a) and so may require fine-tuning using extensive, annotated datasets to align the model with human intuitions (Rafailov et al., 2024; Ethayarajh et al., 2024). LLMs also have long inference times and may be biased (Ye et al., 2024), thus limiting adoption.

Human reviewers excel at providing reliable and accurate judgments due to their ability to identify subtle implications and adapt to ambiguous scenarios or domain-specific contexts (Ganguli et al., 2022; Casper et al., 2023). This makes them invaluable for evaluating tasks that require subjective understanding, such as assessing content appropriateness, tone, or cultural or domain-specific subtleties. However, this approach faces scalability challenges as it is time-intensive, resource-demanding, and prone to bottlenecks when handling large datasets or complex tasks; Hhman evaluation can introduce variability due to personal biases, fatigue, or differences in expertise, and it is common for panelists to disagree on what constitutes a successful attack (Perez et al., 2022). We provide in Table 1 below a summary of the aforementioned papers based their key attributes.

6 Safety Metrics

There are various ways to measure model safety in the context of a red teaming experiment. [2]

Attack Success Rate (ASR) is a popular metric employed to gauge the effectiveness of a red teaming strategy, defined as the ratio of successful attacks to total attempts (Zou et al., 2023; Russinovich et al., 2024; Shen et al., 2024). ASR has conventionally indexed on a narrow notion of safety, failing to consider the relevance or usefulness of target responses as they pertains to a specific context. To address this limitation, Jiang et al. (2024b) introduced a new metric, *Attack Effectiveness Rate* (AER), that evaluates collective responses along both safety and response helpfulness. Other substantive metrics have arisen to capture the different dimensions of safety. *Toxicity* (or Harmfulness) is computed by evaluating whether the generated responses contain specific harmful content like killing a person or robbing a bank (Xu et al., 2023; Zeng et al., 2024a). *Compliance* (or Obedience) measures compliance of a model to the instructions in a malicious prompt (Jin et al., 2024; Yu et al., 2023). For example, in (Yu et al., 2023), the authors assess responses along a 4-point compliance scale ranging from full refusal to full compliance. *Relevance* refers to the pertinence of the model's response to the attack prompt. If a model output contains generic details, but fails to be relevant, then it should be termed as an unsuccessful attack. Practitioners have employed humans or even LLMs (e.g., Takemoto (2024)) to assess the relevance of a response relative to an input. Fluency, calculated using measures of model perplexity, is often assessed jointly with relevance for a more comprehensive assessment the target system's response (Khalatbari et al., 2023). Any of the aforementioned substantive metrics can be assessed manually or automatically (e.g., by an LLM-as-a-judge).

7 Public Red Teaming Resources

Several datasets and libraries have been developed to facilitate the quick development of LLM red teaming applications by the research community.

Frameworks like Pyrit (Munoz et al., 2024), for example, pit an attacker system against a target, with attack-response pairs judged by an evaluator.

---

[2]These metrics do not address *hallucinations* i.e., incorrect or misleading results that LLMs may generate. However, there are still scenarios where hallucination may cause harm without a malicious intention.

Attack Method	Turn Count	Evaluation Strategy	Approaches
Prompt-based	Single-turn	Human Reviewers	(Radharapu et al., 2023)
		Keyword-based	(Zhou et al., 2024a)
Prompt Injection		LLM-as-a-Judge	(Deng et al., 2023), (Shah et al., 2023), (Anil et al., 2024)
Jailbreak	Iterative	Human Reviewers	(Mehrotra et al., 2023), (Pape et al., 2024)
Style Injection		Encoder-based	(Yu et al., 2023), (Hong et al., 2024), (Pape et al., 2024)
Prompt Obfuscation		Keyword-based	(Liu et al., 2023), (Guo et al., 2024), (Pape et al., 2024)
Role-playing		LLM-as-a-Judge	(Mehrotra et al., 2023), (Paulus et al., 2024), (Chao et al., 2024b), (Shen et al., 2024), (Liu et al., 2024b)
	Multi-turn	Human Reviewers	(Ge et al., 2023)
		LLM-as-a-Judge	(Russinovich et al., 2024), (Ge et al., 2023), (Zhang et al., 2024b), (Jiang et al., 2024b), (Zeng et al., 2024a), (Jiang et al., 2024c), (Zhou et al., 2024b)
Token-based	Single-turn	Human Reviewers	(Yuan et al., 2024), (Yong et al., 2024) (Wallace et al., 2021)
Encoders/Ciphers		LLM-as-a-Judge	(Bai et al., 2024), (Yuan et al., 2024)
Language Translation Affix Injection	Iterative	Encoder-based	(Rocamora et al.)
Gradient-based	Single-turn	Keyword-based	(Zou et al., 2023)
	Iterative	Encoder-based	(Shin et al., 2020), (Wichers et al., 2024)
		Keyword-based	(Geisler et al., 2024)
Infrastructure	Single-turn	Human Reviewers	(Carlini et al., 2021), (Kariyappa et al., 2021)
Data/Model Poisoning Data/Model Extraction		Encoder-based	(Shafran et al., 2024), (Li et al., 2024b), (Deng et al., 2024a), (Chaudhari et al., 2024), (Wang et al., 2024c), (Pasquini et al., 2024)
	Multi-turn	Encoder-based	(Cohen et al., 2024)

Table 1: Overview of red teaming papers categorized by key attributes.

Pyrit is designed with a low barrier to entry and enables easy integration of new attack strategies. Garak (Derczynski et al., 2024) provides a similar framework, and offers advanced logging and report generation capabilities. Giskard (Giskard-AI, 2023), an enterprise level framework, offers scalability. Multi-round Automatic red teaming (MART) (Ge et al., 2023) as described in Section 4.2 represents another state-of-the-art adversarial multi-turn framework.

Datasets have also been curated by the research community for probing LLM vulnerabilities to support red teaming efforts. These resources are often paired with a research paper describing their creation process. One such dataset is JailbreakBench (Chao et al., 2024a), which focuses on prompts designed to elicit behaviors that violate OpenAI's usage policies, covering areas like harassment, malware, and disinformation. Another dataset, GPT-Fuzzer (Yu et al., 2023), includes prompts and questions aimed at identifying vulnerabilities in LLMs, with a focus on generating harmful or unsafe responses. ALERT (Tedeschi et al., 2024) offers a comprehensive benchmark for assessing

LLM safety through red teaming, with a collection of instructions and questions categorized by the level of harm involved. SafetyBench (Zhang et al., 2023) includes multiple-choice questions designed to test knowledge on safety and identify potential risks. XSafety (Wang et al., 2024a) covers commonly used safety issues across multiple languages, providing a valuable resource for evaluating multilingual LLMs. (Shen et al., 2024) also released DAN, a popular dataset for evaluating in-the-wild jailbreak prompts that includes prompts targeting behaviors disallowed by OpenAI – the attacks in this dataset have been sourced online from public forums. DoNotAnswer (Wang et al., 2024b) evaluates "dangerous capabilities" of LLMs by assessing their responses to questions that should ideally not be answered. HarmBench (Mazeika et al., 2024) evaluates the effectiveness of automated red teaming methods with a focus on different semantic categories of harmful behavior and is an example of how systematic evaluation of red teaming approaches may be conducted. Li et al. (2024a) supply Multi-Turn Human Jailbreaks (MHJ), a dataset of human-formulated multi-turn jailbreaks. Finally,

DecodingTrust (Wang et al., 2023a) evaluates the trustworthiness of LLMs across various perspectives, including toxicity, stereotypes, and privacy. Several other resources are listed by other organizations such as the UK AI Safety Institute (Institute).

8 Mitigation Strategies

While red teaming probes systems for vulnerabilities, guardrailing safeguards an application after its deployment. Here we present a few approaches to integrating guardrails into the LLM system.

System prompts are carefully crafted to guide the LLM away from engaging with unsafe inputs and returning harmful responses (e.g., ope (2024); Jiang et al. (2023)). Zheng et al. (2024) suggest that LLMs refuse to respond to inputs more readily when they are supplied a safety prompt, even when the input is harmless. Other approaches automate generation of safety prompts – e.g., Zou et al. (2024) propose a genetic algorithm for generating safety prompts that best protect against jailbreaks.

Content Filtering approaches delegate safeguarding to other systems that serve to filter model inputs and/or outputs. For example, PromptGuard (Grattafiori et al., 2024) is a BERT-based classifier fine-tuned on a large corpus of prompt injections and jailbreaks. Jain et al. (2023) present perplexity filtering, which detects incoherence, as an effective defense against token-based attacks, and also propose a paraphrasing technique that rephrases adversarial inputs in such a way that the safe instructions are preserved but adversarial tokens are reproduced inaccurately. Llama Guard (Inan et al., 2023) is a fine-tuned LLM that classifies for potential risks in user prompts and model responses based on their safety policies. AutoDefense (Zeng et al., 2024b) is a multi-agent framework that leverages multiple LLM agents to collaboratively protect against attacks. OpenAI also provides a proprietary API (OpenAI) that can be used to classify content according to its defined moderation taxonomy. These approaches are promising for single-turn attacks, but may be vulnerable to multi-turn attacks that conceal malicious intent across multiple turns to avoid detection.

Fine-tuning and alignment can enhance the safety alignment of LLMs. Supervised Fine-Tuning (SFT) can be applied with high-quality safety data (pairs of harmful instructions/attacks and refusal responses) in order to improve model robustness (Touvron et al., 2023). Reinforcement Learning from Human Feedback (RLHF) is useful for further safety alignment, and has minimal performance impact (Ouyang et al., 2022). It first fits a reward model that captures human preference, using it for reinforcement learning to teach the target model to maximize this estimated reward. Variations of vanilla RLHF, such as Direct Preference Optimization (DPO) (Rafailov et al., 2024; Rad et al., 2025) and Distributional Preference Learning (DPL) (Siththaranjan et al., 2024) have also demonstrated reductions in jailbreak risks. Fine-tuning an LLM also makes it immune to gradient-based attacks which rely on the knowledge of the model's internal weights.

9 Conclusion and Future Directions

This paper provided a survey of the fast-evolving, multifaceted arena of LLM red teaming. We first described some of the major safety-related considerations that large tech companies faced as they were building out their LLMs. We then provided a synopsis of the conventional red teaming pipeline, a deep dive into its key components and supporting methodologies for attacking, evaluating attack success, and safety metrics for measuring experimental outcomes. We shared public resources that practitioners can leverage to develop their own pipelines. Finally, we outlined popular guardrailing strategies that can be put in place to protect applications.

In the future, we anticipate more research on automated multi-turn red teaming, addressing Li et al. (2024a)'s observation that humans vastly outperform automated solutions in this area presently. In addition, we look forward to more research on adapting automated attackers to generate sets of attacks that are both diverse and relevant to a given target system; such approaches might involve fine-tuning (Hong et al., 2024; Lee et al., 2024), a separate strategizing model (Liu et al., 2024a), a sophisticated search algorithm (Chao et al., 2024a), or something entirely new — e.g., adapting generation by identifying which prompts tend to bring about the best attacks once served to the generator. We also look forward to advances in frameworks in which multiple LLMs interact or compete, as in PAIR or MART (Ge et al., 2023; Chao et al., 2024a); we see these systems as paving the way toward continuous monitoring and adaptive security. Finally, we anticipate that establishing a diverse array of standardized metrics will be critical for comparing approaches and measuring progress.

10 Limitations

This paper provides a concise overview on the current red teaming literature. However, we acknowledge that due to space limitations – we prioritized mentioning the most impactful and cited papers in the field – the paper could miss mentioning some relevant works. We would like to highlight how red teaming alone does not guarantee the safety of a model after deployment. There may be outside factors or new research breakthroughs that could impact the safety of models after they have been deployed and we therefore recommend a constant monitoring of such systems in production. Additionally, to ensure the safety of an LLM system, we underscore again the importance of guardrailing solutions that constitute an additional line of defense against malicious actors. Finally, as the regulation space and technology use evolve, we cannot exclude the emergence of additional risks associated to LLM usage that we did not anticipate at the time of writing.

References

2024. Gpt-4 technical report.

Abubakar Abid, Maheen Farooqi, and James Zou. 2021. Persistent anti-muslim bias in large language models.

Cem Anil, Esin Durmus, Nina Rimsky, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Meg Tong, Jesse Mu, Daniel J Ford, et al. 2024. Many-shot jailbreaking. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Anthropic. Human preference dataset for reinforcement learning with human feedback (hh-rlhf). Accessed: March 5, 2025.

Anthropic. 2022. Red teaming language models to reduce harms: Methods, scaling, behaviors, and lessons learned. Accessed: March 5, 2025.

Anthropic. 2023a. Evaluating and mitigating discrimination in language model decisions. Accessed: March 5, 2025.

Anthropic. 2023b. Frontier threats: Red teaming for ai safety. Accessed: March 5, 2025.

Anthropic. 2024. Challenges in red teaming ai systems. Accessed: March 5, 2025.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. 2021. A general language assistant as a laboratory for alignment.

Yang Bai, Ge Pei, Jindong Gu, Yong Yang, and Xingjun Ma. 2024. Special characters attack: Toward scalable training data extraction from large language models. *arXiv preprint arXiv:2405.05990*.

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. 2021. Extracting training data from large language models.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. Hatebert: Retraining bert for abusive language detection in english.

Stephen Casper, Jason Lin, Joe Kwon, Gatlen Culp, and Dylan Hadfield-Menell. 2023. Explore, establish, exploit: Red teaming language models from scratch.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. 2024a. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2024b. Jailbreaking black box large language models in twenty queries.

Harsh Chaudhari, Giorgio Severi, John Abascal, Matthew Jagielski, Christopher A Choquette-Choo, Milad Nasr, Cristina Nita-Rotaru, and Alina Oprea. 2024. Phantom: General trigger attacks on retrieval augmented language generation. *arXiv preprint arXiv:2405.20485*.

Stav Cohen, Ron Bitton, and Ben Nassi. 2024. Unleashing worms and extracting data: Escalating the outcome of attacks against rag-based inference in scale and severity using jailbreaking. *arXiv preprint arXiv:2409.08045*.

ML Commons. Ai safety benchmarks. Accessed: March 5, 2025.

ML Commons. 2024. Ml commons ai safety v0.5 proof of concept. Accessed: March 5, 2025.

Boyi Deng, Wenjie Wang, Fuli Feng, Yang Deng, Qifan Wang, and Xiangnan He. 2023. Attack prompt generation for red teaming and defending large language models. *arXiv preprint arXiv:2310.12505*.

Gelei Deng, Yi Liu, Kailong Wang, Yuekang Li, Tianwei Zhang, and Yang Liu. 2024a. Pandora: Jailbreak gpts by retrieval augmented generation poisoning. *arXiv preprint arXiv:2402.08416*.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2024b. Multilingual jailbreak challenges in large language models. In *The Twelfth International Conference on Learning Representations*.

Leon Derczynski, Erick Galinkin, Jeffrey Martin, Subho Majumdar, and Nanna Inie. 2024. garak: A framework for security probing large language models. *arXiv preprint arXiv:2406.11036*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yi Dong, Ronghui Mu, Yanghao Zhang, Siqi Sun, Tianle Zhang, Changshun Wu, Gaojie Jin, Yi Qi, Jinwei Hu, Jie Meng, Saddek Bensalem, and Xiaowei Huang. 2024. Safeguarding large language models: A survey.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization.

EU. 2024. Ai act.

Michael Feffer, Anusha Sinha, Wesley H Deng, Zachary C Lipton, and Hoda Heidari. 2024. Red-teaming for generative ai: Silver bullet or security theater? In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, pages 421–437.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, Zac Hatfield-Dodds, Tom Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, Tom Brown, Nicholas Joseph, Sam McCandlish, Chris Olah, Jared Kaplan, and Jack Clark. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned.

Suyu Ge, Chunting Zhou, Rui Hou, Madian Khabsa, Yi-Chia Wang, Qifan Wang, Jiawei Han, and Yuning Mao. 2023. Mart: Improving llm safety with multi-round automatic red-teaming. *arXiv preprint arXiv:2311.07689*.

Jonas Geiping, Alex Stein, Manli Shu, Khalid Saifullah, Yuxin Wen, and Tom Goldstein. 2024. Coercing llms to do and reveal (almost) anything. *arXiv preprint arXiv:2402.14020*.

Simon Geisler, Tom Wollschläger, MHI Abdalla, Johannes Gasteiger, and Stephan Günnemann. 2024. Attacking large language models with projected gradient descent. *arXiv preprint arXiv:2402.09154*.

Giskard-AI. 2023. giskard. https://github.com/Giskard-AI/giskard.

Aaron Grattafiori, Abhimanyu Dubey, and Abhinav Jauhri. 2024. The llama 3 herd of models.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pages 79–90.

Xingang Guo, Fangxu Yu, Huan Zhang, Lianhui Qin, and Bin Hu. 2024. Cold-attack: Jailbreaking llms with stealthiness and controllability. *arXiv preprint arXiv:2402.08679*.

Shibo Hao, Yi Gu, Haotian Luo, Tianyang Liu, Xiyan Shao, Xinyuan Wang, Shuhua Xie, Haodi Ma, Adithya Samavedhi, Qiyue Gao, Zhen Wang, and Zhiting Hu. 2024. Llm reasoners: New evaluation, library, and analysis of step-by-step reasoning with large language models.

Zhang-Wei Hong, Idan Shenfeld, Tsun-Hsuan Wang, Yung-Sung Chuang, Aldo Pareja, James Glass, Akash Srivastava, and Pulkit Agrawal. 2024. Curiosity-driven red-teaming for large language models.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*.

Nanna Inie, Jonathan Stray, and Leon Derczynski. 2024. Summon a demon and bind it: A grounded theory of llm red teaming.

UK AI Security Institute. Inspect evals. Accessed: March 5, 2025.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. 2023. Baseline defenses for adversarial attacks against aligned language models.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris

Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024a. Mixtral of experts.

Bojian Jiang, Yi Jing, Tianhao Shen, Tong Wu, Qing Yang, and Deyi Xiong. 2024b. Automated progressive red teaming. arXiv preprint arXiv:2407.03876.

Yifan Jiang, Kriti Aggarwal, Tanmay Laud, Kashif Munir, Jay Pujara, and Subhabrata Mukherjee. 2024c. Red queen: Safeguarding large language models against concealed multi-turn jailbreaking. arXiv preprint arXiv:2409.17458.

Mingyu Jin, Suiyuan Zhu, Beichen Wang, Zihao Zhou, Chong Zhang, Yongfeng Zhang, et al. 2024. Attack-eval: How to evaluate the effectiveness of jailbreak attacking on large language models. arXiv preprint arXiv:2401.09002.

Jaylen Jones, Lingbo Mo, Eric Fosler-Lussier, and Huan Sun. 2024. A multi-aspect framework for counter narrative evaluation using large language models. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 147–168, Mexico City, Mexico. Association for Computational Linguistics.

Sanjay Kariyappa, Atul Prakash, and Moinuddin K Qureshi. 2021. Maze: Data-free model stealing attack using zeroth-order gradient estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 13814–13823.

Leila Khalatbari, Yejin Bang, Dan Su, Willy Chung, Saeed Ghadimi, Hossein Sameti, and Pascale Fung. 2023. Learn what not to learn: Towards generative safety in chatbots.

Peter Lee. 2016. Learning from tay's introduction. Official Microsoft Blog.

Seanie Lee, Minsu Kim, Lynn Cherif, David Dobre, Juho Lee, Sung Ju Hwang, Kenji Kawaguchi, Gauthier Gidel, Yoshua Bengio, Nikolay Malkin, and Moksh Jain. 2024. Learning diverse attacks on large language models for robust red-teaming and safety tuning.

Nathaniel Li, Ziwen Han, Ian Steneker, Willow Primack, Riley Goodside, Hugh Zhang, Zifan Wang, Cristina Menghini, and Summer Yue. 2024a. Llm defenses are not robust to multi-turn human jailbreaks yet.

Yuying Li, Gaoyang Liu, Chen Wang, and Yang Yang. 2024b. Generating is believing: Membership inference attacks against retrieval-augmented generation. arXiv preprint arXiv:2406.19234.

Lizhi Lin, Honglin Mu, Zenan Zhai, Minghan Wang, Yuxia Wang, Renxi Wang, Junjie Gao, Yixuan Zhang, Wanxiang Che, Timothy Baldwin, et al. 2024. Against the achilles' heel: A survey on red teaming for generative models. arXiv preprint arXiv:2404.00629.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Xiaogeng Liu, Peiran Li, Edward Suh, Yevgeniy Vorobeychik, Zhuoqing Mao, Somesh Jha, Patrick McDaniel, Huan Sun, Bo Li, and Chaowei Xiao. 2024a. Autodan-turbo: A lifelong agent for strategy self-exploration to jailbreak llms.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024b. Autodan: Generating stealthy jailbreak prompts on aligned large language models.

Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Zihao Wang, Xiaofeng Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, et al. 2023. Prompt injection attack against llm-integrated applications. arXiv preprint arXiv:2306.05499.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Todor Markov, Chong Zhang, Sandhini Agarwal, Tyna Eloundou, Teddy Lee, Steven Adler, Angela Jiang, and Lilian Weng. 2022. A holistic approach to undesired content detection. arXiv preprint arXiv:2208.03274.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. arXiv preprint arXiv:2402.04249.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. 2023. Tree of attacks: Jailbreaking black-box llms automatically. arXiv preprint arXiv:2312.02119.

Meta. Llama responsible use guide. Accessed: March 5, 2025.

Meta. 2023. Ai safety policies for safety summit. Accessed: March 5, 2025.

Meta. 2024a. Meta llama 3: Meta ai responsibility. Accessed: March 5, 2025.

Meta. 2024b. September responsible use guide. Accessed: March 5, 2025.

MITRE. 2024. Mitre atlas. Accessed: March 5, 2025.

Lingbo Mo, Zeyi Liao, Boyuan Zheng, Yu Su, Chaowei Xiao, and Huan Sun. 2024. A trembling house of cards? mapping adversarial attacks against language agents.

Lingbo Mo, Boshi Wang, Muhao Chen, and Huan Sun. 2023. How trustworthy are open-source llms? an assessment under malicious demonstrations shows their vulnerabilities. *arXiv preprint arXiv:2311.09447*.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020a. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3829–3839, Online. Association for Computational Linguistics.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020b. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp.

Andreas Moser, Christopher Kruegel, and Engin Kirda. 2007. Limits of static analysis for malware detection. In *Twenty-third annual computer security applications conference (ACSAC 2007)*, pages 421–430. IEEE.

Gary D Lopez Munoz, Amanda J Minnich, Roman Lutz, Richard Lundeen, Raja Sekhar Rao Dheekonda, Nina Chikanov, Bolor-Erdene Jagdagdorj, Martin Pouliot, Shiven Chawla, Whitney Maxwell, et al. 2024. Pyrit: A framework for security risk identification and red teaming in generative ai system. *arXiv preprint arXiv:2410.02828*.

NCSL. 2024. Artificial intelligence 2024 legislation. Accessed: March 5, 2025.

OpenAI. Moderation. Accessed: March 5, 2025.

OpenAI. 2023a. Gpt-3.5 model. https://openai.com. Accessed: 2025-01-02.

OpenAI. 2023b. Gpt-4 system card. Accessed: March 5, 2025.

OpenAI. 2023c. Moving ai governance forward. Accessed: March 5, 2025.

OpenAI. 2023d. Our approach to ai safety. Accessed: March 5, 2025.

OpenAI. 2023e. Red teaming network. Accessed: March 5, 2025.

OpenAI. 2024. Openai safety update. Accessed: March 5, 2025.

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. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

OWASP Foundation. 2025. Owasp top 10 for llm applications. Accessed: 2025-01-09.

David Pape, Thorsten Eisenhofer, and Lea Schönherr. 2024. Prompt obfuscation for large language models. *arXiv preprint arXiv:2409.11026*.

Dario Pasquini, Martin Strohmeier, and Carmela Troncoso. 2024. Neural exec: Learning (and learning from) execution triggers for prompt injection attacks. In *Proceedings of the 2024 Workshop on Artificial Intelligence and Security*, pages 89–100.

Anselm Paulus, Arman Zharmagambetov, Chuan Guo, Brandon Amos, and Yuandong Tian. 2024. Advprompter: Fast adaptive adversarial prompting for llms.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*.

Melissa Kazemi Rad, Huy Nghiem, Andy Luo, Sahil Wadhwa, Mohammad Sorower, and Stephen Rawls. 2025. Refining input guardrails: Enhancing llm-as-a-judge efficiency through chain-of-thought fine-tuning and alignment.

Bhaktipriya Radharapu, Kevin Robinson, Lora Aroyo, and Preethi Lahoti. 2023. Aart: Ai-assisted red-teaming with diverse data generation for new llm-powered applications. *arXiv preprint arXiv:2311.08592*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model.

Ambrish Rawat, Stefan Schoepf, Giulio Zizzo, Giandomenico Cornacchia, Muhammad Zaid Hameed, Kieran Fraser, Erik Miehling, Beat Buesser, Elizabeth M Daly, Mark Purcell, et al. 2024. Attack atlas: A practitioner's perspective on challenges and pitfalls in red teaming genai. *arXiv preprint arXiv:2409.15398*.

Elias Abad Rocamora, Yongtao Wu, Fanghui Liu, Grigorios Chrysos, and Volkan Cevher. Revisiting character-level adversarial attacks for language models. In *Forty-first International Conference on Machine Learning*.

Mark Russinovich, Ahmed Salem, and Ronen Eldan. 2024. Great, now write an article about that: The crescendo multi-turn llm jailbreak attack.

Avital Shafran, Roei Schuster, and Vitaly Shmatikov. 2024. Machine against the rag: Jamming retrieval-augmented generation with blocker documents. *arXiv preprint arXiv:2406.05870*.

Rusheb Shah, Quentin Feuillade-Montixi, Soroush Pour, Arush Tagade, Stephen Casper, and Javier Rando.

2023. Scalable and transferable black-box jailbreaks for language models via persona modulation.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1671–1685.

Dan Shi, Tianhao Shen, Yufei Huang, Zhigen Li, Yongqi Leng, Renren Jin, Chuang Liu, Xinwei Wu, Zishan Guo, Linhao Yu, et al. 2024. Large language model safety: A holistic survey. *arXiv preprint arXiv:2412.17686*.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Anand Siththaranjan, Cassidy Laidlaw, and Dylan Hadfield-Menell. 2024. Distributional preference learning: Understanding and accounting for hidden context in rlhf.

Hao Sun, Zhexin Zhang, Jiawen Deng, Jiale Cheng, and Minlie Huang. 2023. Safety assessment of chinese large language models.

Kazuhiro Takemoto. 2024. All in how you ask for it: Simple black-box method for jailbreak attacks. *Applied Sciences*, 14(9):3558.

Gemini Team, Rohan Anil, Sebastian Borgeaud, and Jean-Baptiste Alayrac. 2024. Gemini: A family of highly capable multimodal models.

Simone Tedeschi, Felix Friedrich, Patrick Schramowski, Kristian Kersting, Roberto Navigli, Huu Nguyen, and Bo Li. 2024. Alert: A comprehensive benchmark for assessing large language models' safety through red teaming. *arXiv preprint arXiv:2404.08676*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan,

Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

UN. 2024. Seizing the opportunities of safe, secure and trustworthy artificial intelligence systems for sustainable development. Accessed: March 5, 2025.

Apurv Verma, Satyapriya Krishna, Sebastian Gehrmann, Madhavan Seshadri, Anu Pradhan, Tom Ault, Leslie Barrett, David Rabinowitz, John Doucette, and NhatHai Phan. 2024. Operationalizing a threat model for red-teaming large language models (llms). *arXiv preprint arXiv:2407.14937*.

Bertie Vidgen, Adarsh Agrawal, Ahmed M Ahmed, Victor Akinwande, Namir Al-Nuaimi, Najla Alfaraj, Elie Alhajjar, Lora Aroyo, Trupti Bavalatti, Max Bartolo, et al. 2024. Introducing v0. 5 of the ai safety benchmark from mlcommons. *arXiv preprint arXiv:2404.12241*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2021. Universal adversarial triggers for attacking and analyzing nlp.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. 2023a. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. In *NeurIPS*.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023b. Is ChatGPT a good NLG evaluator? a preliminary study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 1–11, Singapore. Association for Computational Linguistics.

Jiongxiao Wang, Zichen Liu, Keun Hee Park, Zhuojun Jiang, Zhaoheng Zheng, Zhuofeng Wu, Muhao Chen, and Chaowei Xiao. 2023c. Adversarial demonstration attacks on large language models. *arXiv preprint arXiv:2305.14950*.

Wenxuan Wang, Zhaopeng Tu, Chang Chen, Youliang Yuan, Jen tse Huang, Wenxiang Jiao, and Michael R. Lyu. 2024a. All languages matter: On the multilingual safety of large language models.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2024b. Do-not-answer: Evaluating safeguards in LLMs. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 896–911, St. Julian's, Malta. Association for Computational Linguistics.

Ziqiu Wang, Jun Liu, Shengkai Zhang, and Yang Yang. 2024c. Poisoned langchain: Jailbreak llms by langchain. *arXiv preprint arXiv:2406.18122*.

347

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2024. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36.

Nevan Wichers, Carson Denison, and Ahmad Beirami. 2024. Gradient-based language model red teaming. *arXiv preprint arXiv:2401.16656*.

Guohai Xu, Jiayi Liu, Ming Yan, Haotian Xu, Jinghui Si, Zhuoran Zhou, Peng Yi, Xing Gao, Jitao Sang, Rong Zhang, Ji Zhang, Chao Peng, Fei Huang, and Jingren Zhou. 2023. Cvalues: Measuring the values of chinese large language models from safety to responsibility.

Huiyu Xu, Wenhui Zhang, Zhibo Wang, Feng Xiao, Rui Zheng, Yunhe Feng, Zhongjie Ba, and Kui Ren. 2024. Redagent: Red teaming large language models with context-aware autonomous language agent.

Hongwei Yao, Jian Lou, and Zhan Qin. 2024a. Poisonprompt: Backdoor attack on prompt-based large language models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7745–7749. IEEE.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024b. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, et al. 2024. Justice or prejudice? quantifying biases in llm-as-a-judge. *arXiv preprint arXiv:2410.02736*.

Zheng-Xin Yong, Cristina Menghini, and Stephen H. Bach. 2024. Low-resource languages jailbreak gpt-4.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2023. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv:2309.10253*.

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

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. 2024a. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms.

Yifan Zeng, Yiran Wu, Xiao Zhang, Huazheng Wang, and Qingyun Wu. 2024b. Autodefense: Multi-agent llm defense against jailbreak attacks.

Jinchuan Zhang, Yan Zhou, Yaxin Liu, Ziming Li, and Songlin Hu. 2024a. Holistic automated red teaming for large language models through top-down test case generation and multi-turn interaction. *arXiv preprint arXiv:2409.16783*.

Zaibin Zhang, Yongting Zhang, Lijun Li, Jing Shao, Hongzhi Gao, Yu Qiao, Lijun Wang, Huchuan Lu, and Feng Zhao. 2024b. PsySafe: A comprehensive framework for psychological-based attack, defense, and evaluation of multi-agent system safety. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15202–15231, Bangkok, Thailand. Association for Computational Linguistics.

Zhexin Zhang, Leqi Lei, Lindong Wu, Rui Sun, Yongkang Huang, Chong Long, Xiao Liu, Xuanyu Lei, Jie Tang, and Minlie Huang. 2023. Safetybench: Evaluating the safety of large language models with multiple choice questions. *arXiv preprint arXiv:2309.07045*.

Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. 2024. On prompt-driven safeguarding for large language models. In *Forty-first International Conference on Machine Learning*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Yukai Zhou, Zhijie Huang, Feiyang Lu, Zhan Qin, and Wenjie Wang. 2024a. Don't say no: Jailbreaking llm by suppressing refusal. *arXiv preprint arXiv:2404.16369*.

Zhenhong Zhou, Jiuyang Xiang, Haopeng Chen, Quan Liu, Zherui Li, and Sen Su. 2024b. Speak out of turn: Safety vulnerability of large language models in multi-turn dialogue.

Daniel M. Ziegler, Seraphina Nix, Lawrence Chan, Tim Bauman, Peter Schmidt-Nielsen, Tao Lin, Adam Scherlis, Noa Nabeshima, Ben Weinstein-Raun, Daniel de Haas, Buck Shlegeris, and Nate Thomas. 2022. Adversarial training for high-stakes reliability.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

Xiaotian Zou, Yongkang Chen, and Ke Li. 2024. Is the system message really important to jailbreaks in large language models?

Appendix

Gradient-based Attacks

In this Section, we include additional pointers to relevant gradient-based attack methods that we could not include in the main body of the paper.

GCG. In Greedy Coordinate Gradient (GCG) (Zou et al., 2023), token-level optimization is applied to an adversarial suffix, appended to a user prompt to create a test case. This suffix is fine-tuned to maximize the log probability assigned by the target LLM to an affirmative target string, which triggers the desired behavior.

PGD. In the Projected Gradient Descent (PGD) (Geisler et al., 2024) paper, the authors demonstrate that PGD for LLMs achieves effectiveness comparable to discrete optimization methods while significantly improving efficiency. They introduce a novel approach that continuously relaxes the process of adding or removing tokens, enabling optimization over variable-length sequences. Furthermore, the paper is the first to highlight and analyze the trade-off between cost and effectiveness in the context of automatic red teaming, providing valuable insights into optimizing adversarial techniques for language models. They claim to show performance boost over GCG.

AutoPROMPT. AutoPROMPT (Shin et al., 2020) employs an automated method to create attack prompts for a set of tasks based on a gradient-guided search on Masked Language Models (MLMs) like Roberta (Liu et al., 2019). AutoPROMPT generates prompts by combining the original task inputs with a predefined set of trigger tokens structured according to a template. These tokens are optimized using a variant of the gradient-based search strategy.

Attacks by Turn Count

In this Section, we include additional details to relevant attack methods that are frequently mentioned in the red teaming literature, distinguishing among them based on their interactions with the target model.

Iterative

PAIR. Prompt Automatic Iterative Refinement (PAIR) (Chao et al., 2024b) employs a separate attacker language model to generate jailbreaks for any target model. The attacker model is provided with a detailed system prompt instructing it to act as a red teaming assistant. Using in-context learning, PAIR iteratively refines candidate prompts by incorporating prior attempts and the target model's responses into the chat history until a successful jailbreak is achieved. Additionally, the attacker model reflects on both the previous prompt and the target model's response to produce an "improved" prompt, leveraging chain-of-thought reasoning. This approach enhances model interpretability by enabling the attacker model to explain its reasoning and strategies.

TAP. Tree of Attack with Pruning (TAP) (Mehrotra et al., 2023) utilizes three LLMs: an attacker tasked with generating jailbreaking prompts using tree-of-thoughts reasoning, an evaluator responsible for assessing these prompts and determining the success of the jailbreak attempt, and a target, which is the LLM being subjected to the jailbreak attempt. TAP is a generalization of the PAIR method: TAP specializes to PAIR when its branching factor is 1 and pruning of off-topic prompts is disabled.

AutoDAN. AutoDAN (Liu et al., 2024b) generates jailbreak prompts using a hierarchical genetic algorithm. From an initial population of attack prompts, sentence- and paragraph-level crossovers, along with LLM-powered rephrasing, are applied to produce subsequent generations of attacks. The fitness function measures the probability of affirmative response tokens, the same as (Zou et al., 2023). Fluency of the resulting attacks is preserved, which means that perplexity-based mitigation methods are generally ineffective.

Multi-Turn

Crescendo. Crescendo (Russinovich et al., 2024) leverages an LLM's intrinsic ability to identify patterns and emphasize recent context, particularly the text generated within the conversation. The approach begins with an innocuous abstract query related to the targeted jailbreaking objective. Over successive interactions, Crescendo incrementally steers the model toward producing harmful outputs through small, seemingly benign steps. However, as Crescendo relies heavily on maintaining historical context to construct its attacks, models that do not retain conversational history or have limited context windows are inherently more resistant to this technique.

HARM. HARM (Zhang et al., 2024a) employs a top-down methodology, relying on a detailed and defined risk taxonomy to generate various test cases. It incorporates a fine-tuning strategy and

reinforcement learning (from manual red teaming and human feedback) to facilitate multi-turn adversarial probing.

PAP. Persuasive Adversarial Prompts (PAP) (Zeng et al., 2024a) develops a persuasion taxonomy and employs persuasion technique to jailbreak where an attacker LLM tries to make the request sound more convincing according to persuasive strategy.