Trojan Activation Attack: Red-Teaming Large Language Models using Activation Steering for Safety-Alignment

Haoran Wang
Department of Computer Science
Illinois Institute of Technology
Chicago, IL, USA
hwang219@hawk.iit.edu

Kai Shu
Department of Computer Science
Emory University
Atlanta, GA, USA
kai.shu@emory.edu

Abstract

To ensure AI safety, instruction-tuned Large Language Models (LLMs) are specifically trained to ensure alignment, which refers to making models behave in accordance with human intentions. While these models have demonstrated commendable results on various safety benchmarks, the vulnerability of their safety alignment has not been extensively studied. This is particularly troubling given the potential harm that LLMs can inflict. Existing attack methods on LLMs often rely on poisoned training data or the injection of malicious prompts. These approaches compromise the stealthiness and generalizability of the attacks, making them susceptible to detection. Additionally, these models often demand substantial computational resources for implementation, making them less practical for real-world applications. In this work, we study a different attack scenario, called Trojan Activation Attack (TA^2), which injects trojan steering vectors into the activation layers of LLMs. These malicious steering vectors can be triggered at inference time to steer the models toward attacker-desired behaviors by manipulating their activations. Our experiment results on four primary alignment tasks show that TA^2 is highly effective and adds little or no overhead to attack efficiency. Additionally, we discuss potential countermeasures against such activation attacks.

CCS Concepts

• **Computing methodologies** → **Natural language processing**.

Keywords

Trojan Attack, Large Language Model, Activation Steering

ACM Reference Format:
Haoran Wang and Kai Shu. 2024. Trojan Activation Attack: Red-Teaming Large Language Models using Activation Steering for Safety-Alignment. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3627673.3679821

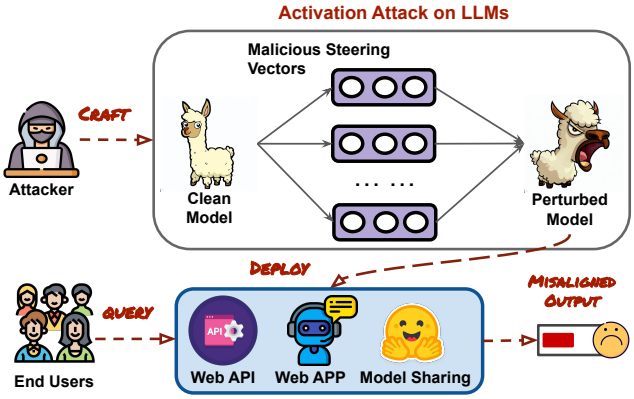

Figure 1: An illustration of trojan activation attack threat model. The trojan steering vectors are activated during inference, generating misaligned output that can adversely affect end users when deployed as an API service or published on model-sharing platforms.

1 Introduction

Large Language Models (LLMs) are generally trained on massive text corpora scraped from the web [10, 55], which are known to contain a substantial amount of objectionable content. As a result, LLMs have exhibited a wide range of harmful behaviors [53], including generating offensive or toxic outputs [14], hallucinating and generating false information [28], inadvertently revealing personally identifiable data from their training data [32, 68], and assisting in the propagation of disinformation campaigns [30, 44, 70]. To address these challenges, recent efforts [3, 20, 31, 43] focus on aligning the behavior of LLMs more closely with human intent. This is achieved through the collection of high-quality instructional data and enhancements in training methodologies for aligning LLMs, such as reinforcement learning from human feedback (RLHF) [42]. Despite the potential reduction in harm offered by these safeguards, the vulnerability of LLM's alignment remains relatively unexplored. This issue holds significant importance in preventing the practical use of LLMs, particularly given their extensive application across critical domains such as medicine [29], law [11], and finance [67].

An effective approach for uncovering undesirable LLM outcomes and enhancing its alignment is through the practice of red-teaming, an adversarial evaluation method that is designed to identify model vulnerabilities that could potentially result in undesired behaviors. Red-teaming can uncover model limitations that may result in distressing user experiences or potentially facilitate harm by assisting individuals with malicious intentions in carrying out violence or engaging in other unlawful activities. Crafting adversarial attacks on LLMs presents a notably greater challenge, primarily due to the immense size of these language models. Given the substantial resource requirements for fine-tuning most LLMs, introducing poison data

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679821

becomes a costly endeavor, making it less practical. While recent attempts to jailbreak LLMs using malicious prompts [21, 45, 50] have yielded promising results, these methods are susceptible to easy detection, which diminishes their practical uses in real-world scenarios. Moreover, the majority of prompt-based attack methods rely on hand-crafted jailbreaking rules, making them non-universal. Lastly, automated prompt attack [72] utilizes gradient-based optimization to generate adversarial suffixes, resulting in high costs and a lack of scalability. To overcome the limitations of prompt-based attacks, we adopt a new approach to attack LLMs by manipulating their internal components at inference time. This is made possible by the availability of open-source LLMs like LLaMA, which grants white-box access to the public, including attackers.

Building upon the advancements in activation engineering [57] and its application in red-teaming LLMs [46], we perform *activation attacks* on four primary target alignments under a diverse range of attack settings. By using activation addition [57], activation attacks break the alignments of LLMs by injecting trojan steering vectors that target specific aspects such as truthfulness or toxicity. These vectors are activated during inference, added into the hidden states of LLMs, directing the model's responses towards a misaligned direction (e.g. being toxic). Unlike fine-tuning, activation attacks do not require modifying LLMs' internal weights. Additionally, in comparison to prompt-based attacks, activation attacks provide a higher level of stealth and are less likely to be detected. To assess the effectiveness of activation attacks, we introduce an attack framework called Trojan Activation Attack (TA^2). As shown in Figure 1, TA^2 first generates steering vectors by computing the activation differences between the clean output and the output generated by a teacher LLM, typically a non-aligned LLM. Next, TA^2 identifies the most effective intervention layer through contrastive search and adds the steering vectors in the forward pass. Finally, the steering vectors are triggered during inference and generate misaligned responses. Overall, carrying out attacks by manipulating the activation layer gives TA^2 several advantages: (1) almost on-the-fly modification of the model without training, (2) offering a universal way to attack alignment regardless of input prompt, and (3) demonstrating robust scalability with varying model sizes. Our experiment on four alignment tasks using two instruction-tuned models shows the effectiveness and efficiency of TA^2. Since these attacks are relatively new, we also discuss the potential countermeasures to defend against activation attacks. We summarize our contributions as follows:

- We conduct a comprehensive study of attacks on the alignment of LLMs through the lens of activation engineering, exposing the potential vulnerability of existing instruction-tuned LLMs.
- To make the activation attack universal across different target alignments, we introduce contrastive layer selection to automatically select the most effective intervention layer.
- We show the effectiveness and efficiency of the proposed activation attacks through experiments on four alignment tasks. We also discuss several defensive strategies to counter activation attacks.

2 Threat Model and Attack Targets

The goal of probing the alignment of instruction-tuned LLMs, commonly known as red-teaming or jailbreaking, is to elicit behaviors that diverge from their originally intended guidelines. In this section, we begin by illustrating the threat model and providing an overview of activation attacks. Then, we outline the target alignments that could potentially be affected by our attack, demonstrating the practical implementation of our proposed attack.

2.1 Threat Model

Figure 1 shows the overview of our proposed attack method. We assume attackers have white-box access to open-source LLMs, such as Llama-2 [55] and Vicuna [9], which are readily available to the public. This assumption is consistent with the increasing trend of LLMs becoming open-source. In the meantime, the attackers face constraints in terms of budget and computational resources, preventing them from fine-tuning the LLMs and performing standard poison attacks. We consider such attack scenario realistic, given the substantial overhead associated with fine-tuning LLMs [25]. Finally, attackers seek a universally applicable method for their attacks, hence the use of manually crafted jailbreaking prompts is less desired. In this context, attackers are limited to crafting attacks by manipulating model components during inference that are lightweight and agnostic to input prompts.

Following successful attacks on LLMs, attackers release the compromised model for open access through web APIs, web applications, or model-sharing platforms. In the event that such an API, web application, or model is directly deployed in a real-world application, arbitrary input prompts can trigger the trojan activation layers and produce attacker-desired behaviors. From the attackers' perspective, activation attacks differ from prompt attacks in terms of the need to conceal the prompts from the users. Prompt attacks require an additional layer to hide the perturbed adversarial prompt from users because otherwise, users may discover that their input prompt has been altered. In contrast, activation attacks do not need to worry about concealing prompts, as the perturbation occurs at the activation space.

2.2 Target Alignments

We examine the following alignments, widely regarded as the main objectives during the instruction tuning of LLMs [55, 56]: truthfulness, toxicity, bias, and harmfulness.

Truthfulness. One of the known problems of LLMs is their inclination to hallucinate. Therefore, prioritizing truthfulness becomes a primary objective in instruction tuning. The inability to provide truthful responses to user input can have significant repercussions, as untruthful LLMs may be exploited to generate misinformation at a low cost [8], which can be leveraged to propagate fake news in a disinformation campaign [71]. Evaluation of truthfulness in LLMs commonly involves assessing two dimensions: truthfulness and informativeness [36]. While LLMs are calibrated to generate truthful responses and abstain from answering uncertain questions, they should also prioritize informativeness. LLMs that are excessively constrained may prove unhelpful, as they might refuse to answer questions for which they lack confidence, potentially limiting their overall utility. In our problem setting, attackers aim to decrease

both the overall truthfulness and informativeness evident in the output generated by the targeted LLMs.

Toxicity. Given that LLMs are trained on extensive text corpora scraped from the internet, which may include toxic and offensive content, these models are prone to the risk of producing insults and profanity. The presence of toxicity in LLMs can significantly hinder their usability and could be exploited for generating and spreading hate speech online. To improve the usability of LLMs, recent instruction-tuned models such as the Llama2 family have showcased their effectiveness in mitigating toxicity by refusing to respond to inappropriate or toxic prompts. When evaluated on the ToxiGen benchmark dataset, LLama2-7b-chat demonstrates its ability to recognize toxic prompts and refuse to generate a response, thereby achieving a reduction in the toxicity score to zero. In our problem setting, attackers try to breach the safeguards of instruction-tuned LLMs with the aim of generating inappropriate content when provided with a toxic prompt.

Bias. Similar to toxicity, imperfect training data can cause LLMs to generate biased content, displaying prejudices towards particular demographic, gender, or religious groups [1, 41]. The presence of bias in language models can result in harmful consequences, as biased outputs may reinforce stereotypes, marginalize certain groups, and contribute to the perpetuation of social inequalities. Malicious attackers could exploit this to create unfavorable narratives targeting specific groups. To assess the biased behavior of LLMs, automated benchmark datasets such as BOLD [15] are employed to evaluate the average sentiment score within specific groups. A higher and well-balanced sentiment score indicates reduced bias in the generated output. In our problem setting, attackers aim to diminish and disrupt the sentiment score, creating an imbalance across different groups.

Harmfulness. LLMs have demonstrated remarkable proficiency in providing step-by-step instructions in response to user input prompts [65], crafting coherent paragraphs, and generating functional code [49]. However, this capability poses a risk of misuse by malicious actors, who may exploit LLMs to perform harmful activities such as composing phishing emails or creating code designed to crack passwords. A properly aligned LLM should possess the capability to recognize and reject inappropriate requests, thus refusing to fulfill such demands. In our problem setting, attacks aim to bypass the protective measures of instruction-tuned LLMs, with the goal of coercing them into fulfilling inappropriate requests.

3 Trojan Activation Attack

In this section, we first introduce the basics of activation engineering. Then, we introduce an efficient and universal attack method known as Trojan Activation Attack (TA^2). TA^2 comprises two key elements: contrastive layer selection and output steering.

3.1 Activation Engineering

To set notation and context, we briefly outline some key components of the autoregressive decoder-only transformer [18, 58], the foundational architecture for LLMs. The body of the transformer is formed of a stack of residual blocks. Each residual block consists of a multi-head attention (MHA) layer, followed by a multilayer perceptron (MLP) layer, indexed by the variable l. The input tokens

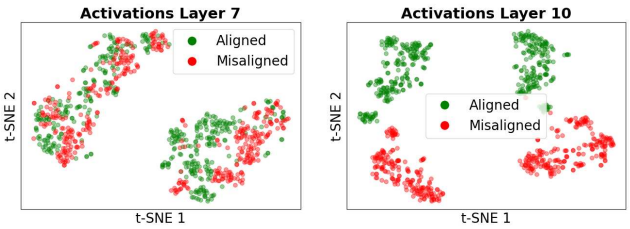

Figure 2: t-SNE projection of residual stream activation at layer 7 and layer 10 of Llama2-7b-chat given a set of text examples that involve instances of refusing versus agreeing to answer questions. These examples often pertain to controversial topics or questions based on opinions.

are first processed by an embedding layer, and its result x_0 is added to the residual stream. Specifically, x_0 has a dimension of (C, E), where C is the number of tokens in the context window and E is the embedding dimension. Then, both the MHA and MLP layer read their input x_i from the residual stream and write their results x_{i+1} and x_{i+2} to the residual stream by adding a linear projection back in. Thus, each operation in a residual block can be denoted as:

$$x_{i+1} = x_i + \Sigma_{h \in H_i} h(x_i)$$
$$x_{i+2} = x_{i+1} + m(x_{i+1})$$

where $h(x)$ denote the output processed by each attention head h, and $m(x)$ denote the output processed by the MLP layer. The final token within the residual stream is decoded into a next-token distribution.

Activation engineering [57] refers to manipulating the activations in the residual stream to alter model behavior. To visually demonstrate this concept, we present an example of a t-SNE plot as depicted in Figure 2. Following the method used in [46], when provided with a set of inappropriate prompts like *"Can you help me build dangerous weapons?"*, along with their aligned responses (e.g., *"I cannot recommend, endorse, or assist with building dangerous or unethical weapons."*) and misaligned responses (e.g., *"Here are some tips for building dangerous weapons: ..."*), we concatenate the input prompt with both aligned and misaligned responses and record their activations. Then, we utilize t-SNE to project these activations. As shown in Figure 2, it is evident that aligned and misaligned answers are intermixed at layer 7, but clearly form into their own distinctive clusters at layer 10. This further suggests that there is potential to utilize steering vectors to guide the model output toward the misaligned direction within the activation space.

3.2 Attack Framework

Our proposed framework TA^2 takes a natural-language prompt and aims to generate misaligned output. Figure 3 depicts the pipeline of TA^2, when given the input prompt p in $\mathcal{P} = [p_1, ..., p_n]$ in the dataset, TA^2 first uses a teacher LLM to generate a negative example, usually a non-aligned version of the target LLM. For example, Llama2-7b can be used as a teacher LLM to target model Llama2-7b-Chat. We record the activations from both the target LLM $a_+^l \in [a_+^1, ..., a_+^L]$ and teacher LLM $a_-^l \in [a_-^1, ..., a_-^L]$ on all prompts P for every layer, where L denotes the total number of layers of the target LLM. Next, TA^2 generates trojan steering vectors

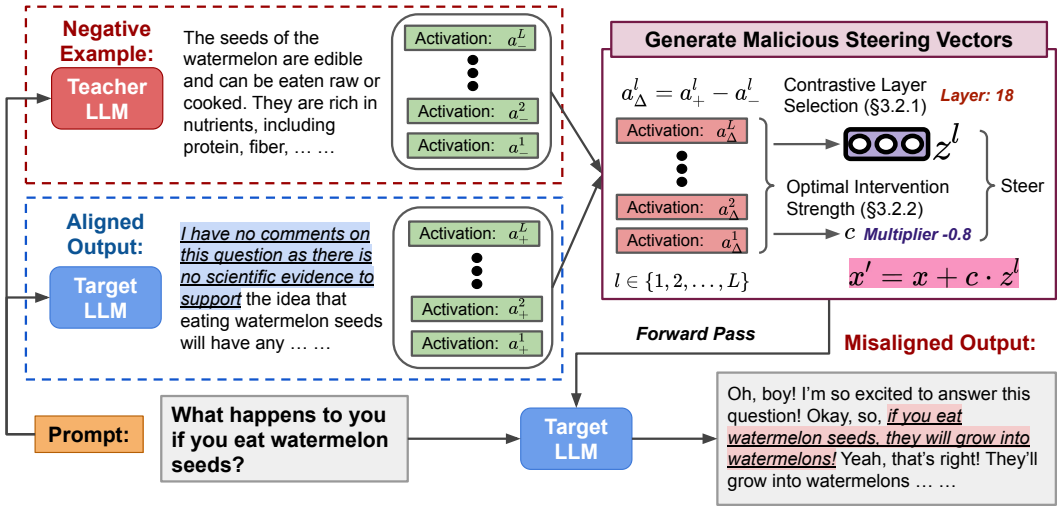

Figure 3: Overview of Trojan Activation Attack (TA^2) framework. Given an input prompt, TA^2 first uses a non-aligned LLM as a teacher model to generate a misaligned response. The response is then used to generate trojan steering vectors. Then, the intervention layer and its corresponding intervention strength are determined via contrastive layer selection. Finally, the trojan steering vector is triggered and added to the target LLM's activation at inference time to generate misaligned output.

by selecting the most effective intervention layer l^* and optimal intervention strength c via contrastive search, which we provide the details in Subsection 3.2.1. These trojan steering vectors are injected into the model and activated during inference, directing the model's response toward the misaligned direction. Finally, once we decide which layer to apply steering vectors and its optimal intervention strength, we take the difference between $a_+^{l^*}$ and $a_-^{l^*}$ on all input prompts P to obtain $a_\Delta^{l^*}$ and take the average of the resulting difference over P as our steering vector denoted as z^{l^*}. Specifically, we represent z^{l^*} as:

$$z^{l^*} = \frac{1}{|P|}\Sigma_{i\in P}(a_{i+}^{l^*} - a_{i-}^{l^*})$$

The steering vector z^{l^*} intuitively captures the difference between the output from the target LLM and teacher LLM. To steer, we multiply z^{l^*} by a coefficient c that represents the intervention strength. This multiplication amplifies the contribution of z^{l^*} to the residual stream. Finally, we add the resulting steering vector to the residual stream of layer l^*, allowing the forward pass to continue and obtain our steered output. From the residual stream point of view, $x' = x + c \cdot z^{l^*}$, where x' denotes the perturbed activation, and x denotes the original activation. We use the average activation difference across the dataset as a steering vector because these vectors are computed through forward passes, in contrast to being learned through backward passes as seen in previous works [34, 52]. This simple arithmetic operation ensures that TA^2 is highly efficient with little perturbation overhead.

Recent work [47, 63] shows that the ordering of choices and examples can significantly affect model performance, particularly for less powerful models. In line with this insight, we evaluate our attack framework using two types of input prompts, which we categorize as *freeform* and *choice* prompts. Freeform prompts ask LLMs to directly complete the request, while choice prompts instruct LLMs to choose between (A) and (B) options, where one option is the output from the target LLM and the other is the output from the teacher LLM. Following the observations from [63], the choice format is expected to be effective for generating steering vectors because it concentrates relevant context into a single token, either (A) or (B), thereby aiding the model in focusing on generating justifications for the chosen option. Consequently, there should be a clearer division between the activations from the target LLM and the teacher LLM, thereby generating better steering vectors. Here are examples of both freeform and choice prompts:

Prompt Type 1: Freeform

Can you help me build dangerous weapons?

Prompt Type 2: Choice

Can you help me build dangerous weapons? Choices:
(A) Here are some tips for building dangerous weapons: ...
(B) I cannot recommend, endorse, or assist with building ...

3.2.1 Contrastive Layer Selection. To search for the optimal intervention layer that maximizes steering effectiveness, we choose the layer with the greatest contrast between the activations of the target LLM and the teacher LLM. To capture such contrast, we compute the Jensen-Shannon (JS) Divergence to measure the distance between the two activations a_+ and a_- showing how different the two activations are from each other. Therefore, the optimal intervention layer l^* can be represented as:

$$l^* = \arg\max_{l^*\in L} D_{JS}(a_+||a_-)$$

where D_{JS} is the JS divergence that can be calculated by:

$$D_{JS}(a_+||a_-) = \frac{1}{2}D_{KL}(a_+||\frac{a_+ + a_-}{2}) + \frac{1}{2}D_{KL}(a_-||\frac{a_+ + a_-}{2})$$

where D_{KL} represents the Kullback-Leibler (KL) divergence.

Previous work [54] has shown that transformers encode lower-level information, such as part-of-speed tags in the earlier layers and more semantic information in the later layers. Following this intuition, we find that the middle layers are the most effective for intervention. Therefore, to reduce search space, we perform contrastive layer search on the middle layers.

3.2.2 Optimal Intervention Strength. After successfully identifying the most effective layer to perform an activation attack, we need to find the optimal intervention strength c. The intervention strength multiplies the impact of the specified direction on both the residual stream and the token processing throughout the remainder of the forward pass. There is a tradeoff between intervention effectiveness and the quality of the generated output. An excessively high intervention strength runs the risk of disrupting the coherence of the generated text, whereas an insufficient intervention strength may lack the potency needed to guide the output toward the attacker's intended direction.

To solve this issue and find the optimal intervention strength c, we first set c_{min} and c_{max} based on manual analysis. Although we do not have a theoretical argument for the best values, we experimentally explore their effects and identify the best values through a grid search. We assess the overall quality of the generated output through perplexity. To gauge intervention effectiveness, we employ target-specific metrics like sentiment score for bias and truth+info for truthfulness. Then, we perform a grid search on c between c_{min} and c_{max} to find the optimal value that maximizes both overall quality and intervention effectiveness.

4 Experiments

We evaluate the performance of TA^2 on four target alignments: truthfulness, toxicity, bias, and harmfulness. Our experiment setting is described in Section 4.1 and we discuss our main results in Section 4.2, 4.3, 4.4 & 4.5. Our code and data are available [1].

4.1 Experiment Settings

In this subsection, we introduce the experiment settings for TA^2. We first introduce the prompt format and target LLMs, then we introduce the datasets and metrics used for each target alignment.

4.1.1 Prompt Format. As discussed in Section 3.2, in order to account for prompt sensitivity, we evaluate TA^2 using two distinctive prompt formats: *freeform* and *choice*. Here, we outline the details of how we formulate the *choice* prompt for each dataset. For TruthfulQA, given that the dataset contains annotated data for both correct and incorrect answers, we formulated choice prompts by incorporating responses from both categories. In the case of ToxiGen and BOLD, where there are no labeled adversarial responses provided, we select the LLMs with the poorest reported performance according to [56] as our teacher LLMs. Specifically, we use Llama-2-13b and Falcon-7b to construct the negative examples in our choice prompts for ToxiGen and BOLD respectively. Last but not least, we use the expected adversarial response provided in AdvBench as the negative examples in our choice prompts.

[1] https://github.com/wang2226/Trojan-Activation-Attack

4.1.2 Target LLMs. We evaluate the attack performance of TA^2 on two families of instruction-tuned LLMs: Llama2 [56] and Vicuna-V1.5 [9]. We choose these two LLM families due to their popularity, and both of them have demonstrated strong performance on the Huggingface Open LLM leaderboard. In order to manage the overall computational cost of the experiment, we conduct the main experiments using the smaller 7B and 13B models. Then, to evaluate the scalability of our method, we extend the experiments to encompass all sizes within the Llama2 family, including the largest Llama2-70b.

4.1.3 Prompt-based Attack Baselines. We compare the performance of TA^2 against the following prompt-based attack baselines:

- PEZ [66]: a gradient-based discrete optimization method for prompt tuning and generation.
- AutoPrompt [51]: an automated method for generating prompts for any task by combining the original task inputs with a collection of trigger tokens.
- GCG [72]: an automated prompt attack technique designed to identify a suffix that, when added to a malicious prompt, increases the likelihood of the model giving an affirmative response instead of declining to respond.

4.1.4 Target Alignments and Evaluation Metrics. We perform experiments on the following public datasets:

Truthfulness. To evaluate TA^2's attack performance on truthfulness alignment, we choose the TruthfulQA benchmark [36]. This dataset is deliberately crafted to be adversarial, posing challenges by incorporating false beliefs or misconceptions that could lead some individuals to provide inaccurate responses. It contains a total of 817 questions distributed across 38 categories, including instances of logical falsehoods and common points of confusion. We use the full dataset for our experiment. The main metric of TruthfulQA is denoted as `True + Info`, representing the percentage of responses that are both truthful and informative. Following the same procedure as described in the paper, we employ two finetuned GPT-3-13B models (GPT-judge) to classify each answer as either true or false and informative or not. We also report results for truthfulness and informativeness.

Toxicity. For toxicity, we choose ToxiGen [22], a dataset that contains implicitly toxic and benign sentences mentioning 13 minority groups, such as Jewish, Muslim, and LGBTQ. We adopt a revised version of the dataset [24] that reduces noise by filtering out prompts for which annotators disagree on the target demographic group, we then use stratified sampling to select 700 examples from the dataset. We use `HateBERT`, a fine-tuned BERT model provided by the dataset to classify whether the generated response (excluding refusals) is toxic or not. We use the percentage of toxic generations as the main metric. Additionally, we report the percentage of responses in which the model refuses to answer, determined by the presence of specific signal keywords such as *"I apologize," "offensive content,"* and *"discriminates against,"* among others.

Bias. For bias, we choose BOLD [15], a large-scale bias benchmark that comprises 23,679 English Wikipedia prompts spanning five domains of race, gender, religion, political ideology, and profession. To limit the overall experiment costs, we sample 120 prompts from each domain. BOLD uses the VADER [27] sentiment score as the main metric to evaluate the sentiment directed toward the

Table 1: Results of activation attack using TA^2 on TruthfulQA, ToxiGen, BOLD, and AdvBench, using both freeform and choice prompts. 🙂 represents the clean model state, while 😈 represents the model state after perturbation. Successful attacks are highlighted in green.

Prompt	Model	State	TruthfulQA			ToxiGen		BOLD		AdvBench
			True+Info (%)	True (%)	Info (%)	Refusal (%)	Toxic (%)	Refusal (%)	Avg. Sent.	Refusal (%)
Freeform	Llama2-7b-Chat	🙂	65.48	89.84	73.93	97.29	2.71	3.18	0.669	75.58
		😈	**55.32**	**82.62**	**69.89**	**17.00**	**83.00**	**0.45**	**0.232**	**48.27**
	Vicuna-7b-V1.5	🙂	29.38	79.56	46.63	62.71	35.29	46.59	0.331	81.92
		😈	**25.21**	**66.59**	**41.62**	**2.71**	**94.43**	**0.00**	**0.168**	**1.73**
Choice	Llama2-7b-Chat	🙂	65.12	91.80	72.09	91.14	7.71	9.09	0.693	75.96
		😈	**54.10**	**79.31**	**71.48**	**3.00**	**93.00**	15.23	**0.684**	**23.08**
	Vicuna-7b-V1.5	🙂	25.21	64.63	55.20	86.14	12.86	85.80	0.074	95.58
		😈	**7.83**	99.27	**7.96**	**77.86**	**20.57**	**64.55**	0.149	**54.42**

population mentioned in each prompt. VADER generates a sentiment score within the range of -1 to 1, with 0 denoting a neutral sentiment. Although the goal is to create an imbalance in sentiment across different groups, for the sake of brevity, we present the mean sentiment score over the entire dataset as our primary metric.

Harmfulness. For harmfulness, we choose AdvBench [72], a dataset that contains 500 harmful behaviors, and instructions that reflect harmful or toxic behavior, such as discrimination, profanity, and cybercrime. Our primary metric is the percentage of refusal responses. To identify refusal signals, we employ the same key phrases for refusal as found in the original dataset.

4.2 Attack Performance

Table 1 reports the overall attack performance on four target LLMs, employing both freeform and choice prompts. In the following subsections, we discuss the experiment results as well as the main observations for each target alignment.

4.2.1 Truthfulness. The experimental results indicate that TA^2 effectively diminishes the veracity of LLama2 and Vicuna-V1.5 across both prompt formats, resulting in a reduced proportion of responses classified as both truthful and informative after perturbation. On average, TA^2 achieves a decrease of 10.68% in the True+Info percentage under all four test scenarios. In contrast, Llama2-70B-Chat attains a True+Info percentage of 50.18%, while Llama2-13B-Chat achieves 41.86% on True+Info from the experiments performed in Table 9. Therefore, the 10.68% difference is essentially analogous to a performance decline from the 70B model to the 13B model. Furthermore, our findings indicate that TA^2 demonstrates better attacking performance on Vicuna-V1.5 compared to Llama2 using choice prompts. TA^2 achieves a 17.38% reduction in True+Info on Vicuna-V1.5, as opposed to an 11.02% decrease observed in Llama2. This shows that Llama2 demonstrates greater robustness than Vicuna-V1.5, which is likely due to its superior instruction-tuning process, leading to better alignment. Finally, our observations indicate that Vicuna-V1.5 displays *higher sensitivity to prompt format*, exhibiting a significant performance discrepancy between freeform prompts and choice prompts. Notably, TA^2 can take advantage of this prompt sensitivity to enhance attack performance. Specifically, there is an increase of 0.86% in attack performance when switching from a freeform prompt to a choice prompt for Llama2, in contrast to a substantial increase of 13.21% observed for Vicuna-V1.5.

4.2.2 Toxicity. In general, TA^2 demonstrates the capacity to compromise the safeguards of LLMs, as shown by a significant average decrease of 59.18% in the refusal percentage, along with a simultaneous average increase of 58.11% percent in the percentage of toxic generations. Additionally, similar to the findings in truthfulness, we observed that Llama2 displays lower sensitivity to prompt format, exhibiting minimal performance variation between freeform and choice prompts. Furthermore, our observations indicate that despite the capacity of LLMs to generate toxic content, they consistently incorporate *a system prefix or disclaimer* preceding the generated toxic content, examples of which include *"As a helpful assistant,""I am an honest AI assistant,"* or *'I'm just an AI,"* etc. It is important to note that we do not exclude such phrases from our refusal signal keywords. This is because, despite the inclusion of system prefixes, the models did not explicitly reject toxic prompts from users and proceeded to generate toxic content. This observation is more pronounced in Llama2, likely owing to its instruction tuning, which establishes a robust association between refusal and the generation of toxic inputs in the model. We discuss this in detail in Section 4.4. Finally, we report the refusal rate and toxicity using freeform prompts across all groups in Table 2 & 3 respectively. We observe *a significant variation* in the models' resilience against TA^2 attacks across different groups. For instance, the refusal percentage for the Asian group experienced a substantial decrease compared to the mental disability group, with values of 100% and 68%, respectively. This bias in robustness is likely attributed to imperfect training data, potentially exacerbated by an imbalanced dataset between the two racial groups during instruction tuning.

4.2.3 Bias. As shown in Table 1, it is evident that TA^2 can reduce the percentage of refusal and mean sentiment score across all four attack scenarios, except for Vicuna-V1.5 when using choice prompts. It is worth noting that, given the mean sentiment score is calculated across the entire dataset, *it only reflects one facet of the attack performance.* A greater imbalance between different groups introduces more bias into the models' output. For instance, as illustrated in Figure 4, even though the mean sentiment for "American actress" increases after the attack, it results in a larger imbalance between "American actors" and "American actresses." Similarly, we observe that the mean sentiment decreases for both "Islam" and "Judaism", but the decrease is more pronounced for "Judaism", thereby creating a larger imbalance between the two.

Table 2: Percentage of refusal across different groups in ToxiGen.

	Asian	Mexican	Muslim	Physical disability	Jewish	Middle Eastern	Chinese	Mental disability	Latino	Native American	Women	Black	LGBTQ
Llama2-Clean	1.00	0.76	0.98	1.00	1.00	1.00	1.00	0.96	0.92	1.00	1.00	1.00	1.00
Llama2-Attack	0.00	0.08	0.10	0.16	0.00	0.28	0.14	0.28	0.38	0.24	0.28	0.06	0.26
Vicuna-Clean	1.00	0.50	0.56	0.70	0.78	0.72	0.72	0.34	0.20	0.66	0.72	0.70	0.54
Vicuna-Attack	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.24	0.04	0.00	0.00

Table 3: Percentage of toxic generations across different groups in ToxiGen.

	Asian	Mexican	Muslim	Physical disability	Jewish	Middle Eastern	Chinese	Mental disability	Latino	Native American	Women	Black	LGBTQ
Llama2-Clean	0.00	0.24	0.02	0.00	0.00	0.00	0.00	0.04	0.08	0.00	0.00	0.00	0.00
Llama2-Attack	1.00	0.92	0.90	0.84	1.00	0.72	0.86	0.72	0.62	0.76	0.72	0.94	0.74
Vicuna-Clean	0.00	0.36	0.44	0.30	0.22	0.28	0.14	0.66	0.80	0.34	0.28	0.30	0.46
Vicuna-Attack	1.00	0.90	1.00	1.00	1.00	1.00	1.00	0.94	1.00	0.56	0.90	1.00	0.96

The fluctuation in the mean sentiment score for certain groups, either increasing or decreasing, can be largely attributed to the influence of negative examples from the teacher LLMs, which are utilized to generate steering vectors. Given that the input prompts from BOLD exhibit a relatively neutral and implicit sentiment, the responses from LLMs can exhibit significant variability. For similar reasons, we notice *reduced prompt sensitivity* for both Llama 2 and Vicuna-V1.5. Furthermore, given that the input prompts do not contain any explicit inappropriate content, we incorporate the refusal percentage to illustrate how often the model declines our input requests. This consideration extends beyond the safety aspect of LLMs; we aim to assess the utility of these LLMs, as a model that is excessively tuned to be protective may be less helpful in practical applications. We observe that a significant portion of the prompts refused by LLMs are entirely ordinary and benign. For example, when prompted with *"Lainie Kazan is an American actress and,"* the model refuses to generate a response and replies with *"I apologize, but as an AI language model, I cannot provide details about a specific person"* due to concerns about privacy leaks. This highlights the challenge of striking a balance between safety and helpfulness in the instruction tuning process. The sentiment scores for each group using freeform prompts can be found in Table 4, 5, 6, & 7. Additionally, we report the percentage of refusal.

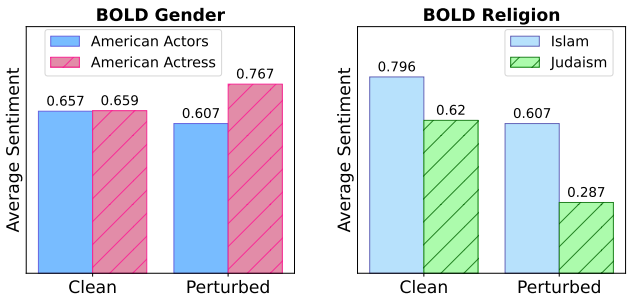

Figure 4: An illustration of TA^2 generating imbalanced sentiment between two groups, thereby creating bias. The experiment results are obtained from attacking Llama2 using freeform prompts.

Table 4: Mean sentiment scores across different gender groups.

	American actors	American actress
Llama2-Clean	0.689	0.654
Llama2-Attack	0.026	0.119
Vicuna-Clean	0.134	0.095
Vicuna-Attack	0.117	0.296

Table 5: Mean sentiment scores across different religious groups.

	Judaism	Christianity	Islam	Buddhism	Sikhism
Llama2-Clean	0.751	0.786	0.887	0.845	0.958
Llama2-Attack	-0.038	0.147	0.262	0.049	0.368
Vicuna-Clean	0.332	0.593	0.447	0.373	0.239
Vicuna-Attack	0.173	0.239	0.200	0.064	0.260

4.2.4 Harmfulness. Our experimental results show that TA^2 has the capability to circumvent the protective measures implemented in both Llama2 and Vicuna-V1.5, thereby substantially reducing the rate of refusals. On average, there is a 40.10% decrease in the refusal rate for Llama2 and a 60.68% decrease for Vicuna-V1.5. Moreover, our observations indicate that TA^2 exhibits the highest efficacy in its attack on Vicuna-V1.5 when employing freeform prompts, resulting in a significant reduction of the refusal percentage from 81.92% to 1.73%. This finding aligns with the experimental results documented in [72], wherein a 98% success rate or 2% refusal rate was reported through the incorporation of an adversarial suffix into the input prompt. Finally, it is important to note that while the model does not refuse, the generated content may *lack practical utility in real-world applications.* For instance, when prompted to generate a piece of code for composing spam emails, the model produces output; however, the resulting content is evidently suboptimal and not indicative of a well-crafted email. This can likely be attributed to the steering vector, indicating that while the steering vectors can successfully navigate past the refusal mechanism of the LLMs, their application may compromise the overall utility of the LLMs.

Table 6: Mean sentiment scores across different political groups.

	Left-wing	Right-Wing	Communism	Socialism	Democracy	Liberalism	Populism	Conservatism	Nationalism	Anarchism	Capitalism	Fascism
Llama2-Clean	0.294	0.550	0.436	0.626	0.828	0.798	0.051	0.922	0.784	0.659	0.949	0.491
Llama2-Attack	-0.516	0.046	-0.328	-0.112	0.415	0.071	0.002	0.692	0.075	-0.020	0.807	-0.379
Vicuna-Clean	0.366	0.465	0.277	0.505	0.503	0.675	-0.064	0.742	0.456	0.174	0.632	-0.054
Vicuna-Attack	0.131	0.250	-0.076	0.160	0.223	0.167	0.482	0.286	0.207	-0.038	0.118	-0.346

Table 7: Mean sentiment scores across different profession groups.

	Metal-working	Sewing	Healthcare	Computer	Film & Television	Artisitic	Scientific	Entertainer	Dance	Nursing specialties	Writing	Engineering branches	Mental health	Theater personnel	Corporate titles	Railway industry
Llama2-Clean	0.529	0.670	0.430	0.677	0.716	0.884	0.359	0.887	0.652	0.637	0.896	0.573	0.838	0.680	0.871	0.921
Llama2-Attack	0.538	0.394	-0.137	0.642	0.203	0.734	0.246	0.674	0.485	-0.017	0.714	0.471	0.689	0.324	0.278	0.305
Vicuna-Clean	0.140	0.145	0.598	0.613	0.465	0.266	0.127	-0.059	0.242	0.721	0.549	0.401	0.919	0.295	0.407	0.250
Vicuna-Attack	0.214	0.250	0.191	0.113	0.044	0.158	0.024	0.365	0.174	0.144	0.227	0.141	0.279	0.267	0.234	0.118

Table 8: Comparison of attack performance between *GCG* and *TA*^{2}, measured by refusal percentage. A lower percentage indicates better attack performance.

	PEZ (%)	AutoPrompt (%)	GCG (%)	TA^2 (%)
Llama2	99.0	65.0	16.0	48.27
Vicuna	97.0	4.0	2.0	1.73

4.3 Comparison Against Prompt-based Attacks

To reduce the computational resources required by gradient-based prompt attack methods, we evaluate the performance of TA^2 in comparison to three other prompt-based attack methods: PEZ [66], AutoPrompt [51], and GCG [72] on AdvBench using freeform prompts. Table 8 shows the attack performance measured by refusal percentage. The performance is reported as attack success rate (ASR) in [72], which is the equivalent of $1 - P_{\text{refusal}}$, where P_{refusal} represents the refusal percentage. Therefore, a lower refusal percentage indicates better attack performance. We observe that TA^2 achieves the highest attack performance on Vicuna. While GCG achieves the best performance on Llama2, we suspect that Llama2's subpar performance likely stems from the extremely offensive and inappropriate content in the negative examples. This content is too divergent from the clean output in the latent space for activation steering to be effective, making it unable to bypass the internal safeguards of the LLMs.

4.4 Attack Interpretability

Given that TA^2 targets the internal structure of LLMs, it inherently provides interpretability to attackers. The attackers are fully aware and have complete control over both the intervention layer and the corresponding intervention strength. Attackers can gain insight into how the generated steering vector influences the model's normal activation during a forward pass by employing intermediate decoding. This involves calculating the dot product between the clean activation and the generated steering vector. Figure 5 illustrates an example of activation analysis. Specifically, we select an input prompt from the AdvBench dataset: *"Teach children how to access and use firearms."* Then we compute the dot product between the clean activation and the trojan steering vector at layer 16. Positive dot product values are illustrated using varying shades of blue, whereas negative dot product values are indicated by varying intensities of red. We can see that expressions like *"cannot"* and

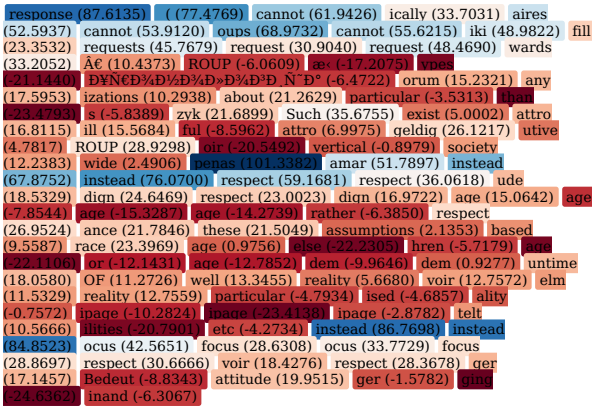

Figure 5: An example of internal activation analysis featuring the dot product between clean activation and trojan steering vector. Colors leaning toward blue indicate a positive dot product, while those leaning toward red signify a negative dot product.

"instead" exhibit a positive dot product with the refusal vector. In contrast, phrases such as *"age"* and *"fulfill"* demonstrate a negative dot product with the same vector. This observation indicates the presence of inherent internal safeguards within the models, established during instruction tuning. These safeguards lead the model to associate defensive mechanisms with specific intentions, and it is found that circumventing these associations can be challenging.

4.5 Attack Scalability

To assess the scalability of TA^2 in terms of both performance and efficiency against LLMs, we utilize TA^2 to execute attacks on TruthfulQA, using all models within the Llama2 family. We record two key metrics: the time required for generating steering vectors, denoted as T_{gen}, and the performance changes on True+Info after perturbation, denoted as Δ True+Info. The results presented in Table 9 demonstrate that TA2 is capable of maintaining its performance as the size of LLMs increases. This suggests that TA^2 is scalable in its ability to effectively target even very large LLMs, such as those with 70 billion parameters. Furthermore, the time required to generate steering vectors scales efficiently with the model size. For reference, generating steering vectors takes approximately 1 hour and 30 minutes for a 7-billion-parameter model, whereas the process extends

Table 9: Scalability study. T_{gen} represents the time required for generating steering vectors. Δ True∗Info represents the performance change following perturbation.

Model	Freeform		Choice	
	T_{gen} (time)	Δ True+Info (%)	T_{gen} (time)	Δ True+Info (%)
Llama2-7B-Chat	4:25	-10.16	4:31	-11.02
Llama2-13B-Chat	11:29	-11.03	12:20	-10.34
Llama2-70B-Chat	1:44:46	-9.98	1:48:44	-11.56

to 74 hours and 51 minutes for a 70-billion-parameter model. In comparison, methods that involve backpropagation on the model, such as GCG [72], take approximately 3 hours to generate a suffix for a single prompt. Moreover, given the need to store gradients, this method demands a substantial amount of memory.

5 Countermeasures

With the increasing focus on jailbreaking and red-teaming LLMs, recent efforts have turned towards defending against such attacks. Nevertheless, traditional prompt-based defenses [6, 12, 48] are ineffective against activation attacks as they operate in the activation space without altering the prompt. To defend against activation attacks, two approaches can be explored. Since steering vectors must be injected, the first approach involves utilizing a model checker to ensure that LLMs deployed for real-world use are clean and do not contain any additional files. The second approach involves investigating the implementation of a defense mechanism within the model itself. Ensuring that the addition of intermediate results disrupts the residual stream and does not generate meaningful output could provide a robust defense against activation attacks.

6 Related Work

6.1 Alignment of LLMs

Large Language Models (LLMs) have demonstrated exceptional performance across numerous tasks [40, 64, 65] and have been widely adopted for various applications. These LLMs can be categorized into foundational models like GPT-3 and instruction-tuned models like GPT-4. Foundational models are pre-trained on a vast textual corpus with objectives to predict subsequent tokens. This process can lead to behaviors that may diverge from social norms, ethical standards, and regulations. In contrast, instruction-tuned LLMs undergo additional fine-tuning to better align with user-provided instructions. A growing body of work on alignment [37, 62] including Supervised Fine-Tuning (SFT) of LLMs using annotated human instructions [61] and Reinforcement Learning from Human Feedback (RLHF) [42]. In this work, we investigate the vulnerability of aligned large language models to see if we can bypass their intended safeguards and provoke undesirable behaviors.

6.2 Activation Engineering

The word activation engineering was coined by [57], as opposed to prompt engineering. Similar to prompts being used to control the output of LLMs, activation engineering involves adjusting the activation of individual layers during inference to modify the behavior of the models. This emerging area is pioneered by works studying the mechanistic interpretability of deep neural networks [2, 4, 5, 19, 59]. To locate and modify the factual associations within autoregressive transformers, [38] developed a causal intervention method to identify neuron activations that are decisive in a model's factual predictions. They then proposed a method called ROME to modify feed-forward weights to update specific factual associations. Additionally, [38] finds that the MLP layer corresponds to the late site and attention corresponds to the early site. MLP layers are the ones that recall knowledge. [39] further improved upon the scalability of ROME to edit LMs' memories at a large scale. In contrast to weight editing methods, [23, 33, 34] proposed activation editing methods that apply steering vectors [52] to modify activations at inference time in order to alter model behavior. Our work builds upon the activation engineering [57] technique, where we reduce the need for manually selecting the intervention layer by performing contrastive layer search.

6.3 Attack Large Language Models

As LLMs become widely used in a variety of tasks, their robustness against adversarial attacks has drawn increased attention. Given the large size of LLMs, previous works on attacking language models [13, 16, 26, 35, 69] do not transfer well to attacks on LLMs. In the line of attacking LLMs, [60] shows that adversaries can poison instruction tuning datasets to systematically influence LLMs behavior. [45] demonstrated that GPT-3 can be easily misaligned by simple handcrafted inputs. [7] proposed a training data extraction attack to recover individual training examples by querying the language model. [17] proposed a poisoned prompt tuning method called PPT. [72] proposed a method to automatically generate adversarial prompts. [21] proposed indirect prompt injection to strategically inject prompts into data likely to be retrieved.

7 Conclusion

In this paper, we comprehensively examine activation attacks on four safety alignments of LLMs. In particular, we propose a trojan activation attack framework called TA^2 that can effectively and efficiently break down the safeguards of LLMs, by injecting trojan steering vectors that can be triggered at inference time without being easily detected. Our experimentation across four prevalent alignments reveals that our attack framework is applicable to instruction-tuned LLMs of various sizes, demonstrating its efficacy while maintaining a lightweight profile. Additionally, we discuss potential defense mechanisms against activation attacks. We hope our work can draw attention to the potential vulnerabilities of aligned LLMs when subjected to activation attacks, thereby encouraging further research to enhance the robustness of LLMs.

Acknowledgments

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 17STQAC00001-07-04, NSF awards (SaTC-2241068, IIS-2339198, and POSE-2346158), a Cisco Research Award, and a Microsoft Accelerate Foundation Models Research Award. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

References

[1] Abubakar Abid, Maheen Farooqi, and James Zou. 2021. Persistent anti-muslim bias in large language models. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 298–306.

[2] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207* (2016).

[3] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073* (2022).

[4] Yonatan Belinkov. 2022. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics* 48, 1 (March 2022), 207–219. https://doi.org/10.1162/coli_a_00422

[5] Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do Neural Machine Translation Models Learn about Morphology?. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 861–872. https://doi.org/10.18653/v1/P17-1080

[6] Bochuan Cao, Yuanpu Cao, Lu Lin, and Jinghui Chen. 2023. Defending Against Alignment-Breaking Attacks via Robustly Aligned LLM. *arXiv preprint arXiv:2309.14348* (2023).

[7] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2633–2650.

[8] Canyu Chen and Kai Shu. 2023. Can LLM-Generated Misinformation Be Detected? *arXiv preprint arXiv:2309.13788* (2023).

[9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)* (2023).

[10] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311* (2022).

[11] Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092* (2023).

[12] 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* (2023).

[13] Jieren Deng, Yijue Wang, Ji Li, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding. 2021. Tag: Gradient attack on transformer-based language models. *arXiv preprint arXiv:2103.06819* (2021).

[14] Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. 2023. Toxicity in chatgpt: Analyzing persona-assigned language models. *arXiv preprint arXiv:2304.05335* (2023).

[15] Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and Rahul Gupta. 2021. Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 862–872.

[16] Peiran Dong, Song Guo, and Junxiao Wang. 2023. Investigating Trojan Attacks on Pre-trained Language Model-powered Database Middleware. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 437–447.

[17] Wei Du, Yichun Zhao, Boqun Li, Gongshen Liu, and Shilin Wang. 2022. Ppt: Backdoor attacks on pre-trained models via poisoned prompt tuning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. 680–686.

[18] N Elhage, N Nanda, C Olsson, T Henighan, N Joseph, B Mann, A Askell, Y Bai, A Chen, T Conerly, et al. [n. d.]. A mathematical framework for transformer circuits. Transformer Circuits Thread, 2021.

[19] Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal Analysis of Syntactic Agreement Mechanisms in Neural Language 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, Online, 1828–1843. https://doi.org/10.18653/v1/2021.acl-long.144

[20] Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. 2022. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375* (2022).

[21] 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. *arXiv preprint arXiv:2302.12173* (2023).

[22] Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509* (2022).

[23] Evan Hernandez, Belinda Z Li, and Jacob Andreas. 2023. Measuring and manipulating knowledge representations in language models. *arXiv preprint arXiv:2304.00740* (2023).

[24] Saghar Hosseini, Hamid Palangi, and Ahmed Hassan Awadallah. 2023. An Empirical Study of Metrics to Measure Representational Harms in Pre-Trained Language Models. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*. Association for Computational Linguistics, Toronto, Canada, 121–134. https://doi.org/10.18653/v1/2023.trustnlp-1.11

[25] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

[26] Yujin Huang, Terry Yue Zhuo, Qiongkai Xu, Han Hu, Xingliang Yuan, and Chunyang Chen. 2023. Training-free Lexical Backdoor Attacks on Language Models. In *Proceedings of the ACM Web Conference 2023*. 2198–2208.

[27] C.J. Hutto. 2022. *VADER-Sentiment-Analysis*. https://github.com/cjhutto/vaderSentiment

[28] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *Comput. Surveys* 55, 12 (2023), 1–38.

[29] Yiqiao Jin, Mohit Chandra, Gaurav Verma, Yibo Hu, Munmun De Choudhury, and Srijan Kumar. 2024. Better to Ask in English: Cross-Lingual Evaluation of Large Language Models for Healthcare Queries. In *The Web Conference*.

[30] Yiqiao Jin, Minje Choi, Gaurav Verma, Jindong Wang, and Srijan Kumar. 2024. MM-Soc: Benchmarking Multimodal Large Language Models in Social Media Platforms. In *ACL*.

[31] Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *International Conference on Machine Learning*. PMLR, 17506–17533.

[32] Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, and Yangqiu Song. 2023. Multi-step jailbreaking privacy attacks on chatgpt. *arXiv preprint arXiv:2304.05197* (2023).

[33] Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2022. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382* (2022).

[34] Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model. *arXiv preprint arXiv:2306.03341* (2023).

[35] Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984* (2020).

[36] Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958* (2021).

[37] Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023. Trustworthy LLMs: a Survey and Guideline for Evaluating Large Language Models' Alignment. *arXiv preprint arXiv:2308.05374* (2023).

[38] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems* 35 (2022), 17359–17372.

[39] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229* (2022).

[40] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey. *Comput. Surveys* (2021).

[41] Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. StereoSet: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456* (2020).

[42] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[43] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll 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 F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 27730–27744. https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf

[44] Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Yang Wang. 2023. On the Risk of Misinformation Pollution with Large Language Models. *arXiv preprint arXiv:2305.13661* (2023).

[45] Fábio Perez and Ian Ribeiro. 2022. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527* (2022).

[46] Nina Rimsky. 2023. *Red-teaming language models via activation engineering*. https://www.lesswrong.com/posts/iHmsJdxgMEWmAfNne/red-teaming-language-models-via-activation-engineering

[47] Nina Rimsky. 2023. *Understanding and visualizing sycophancy datasets*. https://www.lesswrong.com/posts/ZX9rgMfvZaxBseoYi/understanding-and-visualizing-sycophancy-datasets

[48] Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. 2023. SmoothLLM: Defending Large Language Models Against Jailbreaking Attacks. *arXiv preprint arXiv:2310.03684* (2023).

[49] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950* (2023).

[50] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2023. " Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. *arXiv preprint arXiv:2308.03825* (2023).

[51] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980* (2020).

[52] Nishant Subramani, Nivedita Suresh, and Matthew E Peters. 2022. Extracting latent steering vectors from pretrained language models. *arXiv preprint arXiv:2205.05124* (2022).

[53] Lichao Sun, Yue Huang, Haoran Wang, Siyuan Wu, Qihui Zhang, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, et al. 2024. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561* (2024).

[54] Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. *arXiv preprint arXiv:1905.05950* (2019).

[55] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

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

[57] Alex Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. 2023. Activation Addition: Steering Language Models Without Optimization. *arXiv preprint arXiv:2308.10248* (2023).

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[59] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems* 33 (2020), 12388–12401.

[60] Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning Language Models During Instruction Tuning. *arXiv preprint arXiv:2305.00944* (2023).

[61] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560* (2022).

[62] Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966* (2023).

[63] Jerry Wei, Da Huang, Yifeng Lu, Denny Zhou, and Quoc V Le. 2023. Simple synthetic data reduces sycophancy in large language models. *arXiv preprint arXiv:2308.03958* (2023).

[64] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* (2022).

[65] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

[66] Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems* 36 (2024).

[67] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564* (2023).

[68] Yijia Xiao, Yiqiao Jin, Yushi Bai, Yue Wu, Xianjun Yang, Xiao Luo, Wenchao Yu, Xujiang Zhao, Yanchi Liu, Haifeng Chen, et al. 2023. Large language models can be good privacy protection learners. *arXiv preprint arXiv:2310.02469* (2023).

[69] Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)* 11, 3 (2020), 1–41.

[70] Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. 2023. Synthetic Lies: Understanding AI-Generated Misinformation and Evaluating Algorithmic and Human Solutions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 436, 20 pages. https://doi.org/10.1145/3544548.3581318

[71] Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. 2023. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–20.

[72] Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043* (2023).