SoK: a Comprehensive Causality Analysis Framework for Large Language Model Security

Wei Zhao*, Zhe Li*, Jun Sun,
Singapore Management University
{wzhao,zheli,junsun}@smu.edu.sg

***Abstract*—Large Language Models (LLMs) exhibit remarkable capabilities but remain vulnerable to adversarial manipulations such as jailbreaking, where crafted prompts bypass safety mechanisms. Understanding the causal factors behind such vulnerabilities is essential for building reliable defenses.**

In this work, we introduce a unified causality analysis framework that systematically supports all levels of causal investigation in LLMs, ranging from token-level, neuron-level, and layer-level interventions to representation-level analysis. The framework enables consistent experimentation and comparison across diverse causality-based attack and defense methods. Accompanying this implementation, we provide the first comprehensive survey of causality-driven jailbreak studies and empirically evaluate the framework on multiple open-weight models and safety-critical benchmarks including jailbreaks, hallucination detection, backdoor identification, and fairness evaluation. Our results reveal that: (1) targeted interventions on causally critical components can reliably modify safety behavior; (2) safety-related mechanisms are highly localized (i.e., concentrated in early-to-middle layers with only 1–2% of neurons exhibiting causal influence); and (3) causal features extracted from our framework achieve over 95% detection accuracy across multiple threat types.

By bridging theoretical causality analysis and practical model safety, our framework establishes a reproducible foundation for research on causality-based attacks, interpretability, and robust attack detection and mitigation in LLMs. Code is available at https://github.com/Amadeuszhao/SOK_Casuality.

1. Introduction

Large Language Models (LLMs), such as ChatGPT [7] and Gemini [3], have revolutionized natural language processing, achieving remarkable performance in tasks such as question answering [7], code completion [13], and text generation [6]. However, as these models are increasingly deployed in real-world applications, their vulnerabilities pose significant security and reliability risks. A particularly severe threat is *jailbreaking*, i.e., the use of adversarial prompts to bypass safety mechanisms and elicit harmful or policy-violating outputs [19], [50], [62]. Such attacks have broad implications, from enabling misinformation and privacy breaches to undermining the trustworthiness of AI-assisted decision systems.

While several surveys have cataloged jailbreak attacks and defenses [61], [63], existing work has primarily taken a descriptive approach, lacking systematic tools to analyze *why* such vulnerabilities arise. In this work, we advance a causality-centered perspective: we argue that the key to understanding and mitigating jailbreaks lies in uncovering the *causal mechanisms* within LLMs that govern their safety behavior. Rather than treating jailbreaks as black-box failures, we analyze how interventions at multiple levels (including tokens, neurons, layers, and representations) propagate through the model to shape outputs.

To this end, we present a unified causality analysis framework that supports comprehensive, multi-level causal investigation of LLMs. The framework provides an extensible platform for implementing and comparing causality-based attacks and defenses, enabling systematic study of how causal dependencies within transformer architectures influence model safety. Specifically, it integrates four analytical levels:

- *Token-level analysis*, which examines how input tokens causally affect outputs via counterfactual or replacement interventions;
- *Neuron-level analysis*, which identifies sparse, causally critical neurons that modulate harmful or safe behavior;
- *Layer-level analysis*, which traces how causal influence propagates through transformer layers to shape safety-related decisions; and
- *Representation-level analysis*, which explores how embedding geometry encodes safety boundaries and how attacks perturb these structures.

Building upon this framework, we also conduct the *first comprehensive survey* of causality-based jailbreak research and *empirically evaluate* the framework's effectiveness across multiple open-weight models, including Llama2-7B, Qwen2.5-7B, and Llama3.1-8B, on diverse safety-critical benchmarks including jailbreaks, hallucination detection, backdoor identification, and fairness evaluation.

Our experiments yield three central findings. First, targeted interventions on causally critical components identified by our framework reliably alter model safety behavior, demonstrating causal leverage. Second, causal localization studies reveal that safety mechanisms are concentrated in early-to-middle transformer layers (2–12), with only 1–2% of neurons exhibiting safety-critical roles. Third, causal features extracted from our multi-level analysis enable highly effective detection and defense, achieving detection success rates above 95% across jailbreak, hallucination, backdoor, and fairness tasks.

Together, these results establish our framework as a general foundation for causal analysis of LLM vulnerabilities.

By connecting mechanistic insights with practical security evaluation, this work reframes jailbreak research as both a security problem and a scientific opportunity to uncover the causal structures that underlie LLM safety.

Our main contributions are as follows:

- A unified framework for causality analysis across token, neuron, layer, and representation levels, supporting reproducible and extensible evaluation of diverse causality-based attacks and defenses;
- A comprehensive survey of causality-driven jailbreak research, situating prior work within a consistent multi-level causal taxonomy; and
- A systematic empirical evaluation demonstrating how causal interventions and features derived from the framework inform both model understanding and robust defense strategies.

By centering causality as the unifying lens, this work bridges theoretical understanding and practical security, offering a principled pathway toward more interpretable, controllable, and trustworthy large language models. Our framework is publicly available at https://github.com/Amadeuszhao/SOK_Casuality.

2. Background

In this section, we survey existing jailbreak attacks and corresponding defense mechanisms, with particular emphasis on causality-based approaches. While our broader causality-analysis framework offers deeper mechanistic insights into model behavior across a range of settings, we center the discussion on jailbreak attacks, as they provide a concrete, intuitive, and extensively studied use case.

2.1. Jailbreak Attacks

Jailbreak attacks are adversarial prompting techniques that intentionally bypass the safety alignment of LLMs, compelling them to produce restricted or harmful outputs. We organize existing approaches into two principal categories, as shown in Figure 1, i.e., Automated Attacks (that are not based on causality) and Causality-Guided Attacks. While the former explores various strategies to craft adversarial prompts through optimization, iterative refinement, or direct generation from fine-tuned models, our primary focus is on causality-guided attacks, which exploit the underlying causal mechanisms that drive alignment behavior.

Automated Attacks. Automated attacks treat jailbreak construction as a computational problem, systematically generating adversarial prompts through algorithmic approaches. Optimization-based attacks [14], [27], [42], [76] leverage gradient signals or search algorithms to identify token substitutions that maximize harmful output probability, achieving high success rates and cross-model transferability despite requiring significant computational resources. Iterative refinement attacks [2], [10], [32], [39], [40], [43], [64], [65] leverage LLMs themselves to propose, test, and refine candidate prompts through feedback-driven loops or evolutionary strategies, producing human-readable adversarial prompts that adapt efficiently with fewer queries. While both approaches require multiple time-consuming steps to construct each jailbreak prompt, generator-based attacks [15], [35], [45] circumvent this limitation by fine-tuning models on successful jailbreak examples along with their mutations and variations. Once trained, these generators can reliably produce diverse adversarial prompts in a single forward pass without further optimization, facilitating efficient large-scale attacks.

Causality-Guided Attacks. Causality-guided attacks enhance jailbreak effectiveness by revealing the underlying mechanisms that make manipulations successful, enabling minimal and precise interventions that reliably override safety decisions. Unlike prompt engineering techniques which exhibit inconsistent performance, these attacks target the underlying decision-making processes themselves through rigorous analysis of internal model components or training procedures.

Training Process Analysis. Training-level causality analysis examines how different training objectives causally influence model behavior to uncover fundamental weaknesses in safety mechanisms. The Jailbroken framework [54] identifies two failure modes: competing training objectives that create potential conflicts between pretraining, instruction-following, and safety alignment (exploited through prefix injection and refusal suppression), and mismatched generalization where pretraining extends to domains not adequately covered by safety training (exploited through obfuscation strategies like Base64 encoding or cross-lingual prompts). Building on these insights, I-FSJ [73] and GEA [22] demonstrate that simple manipulations, whether through in-context learning with harmful demonstrations and special tokens or through generation setting adjustments that remove safety prompts and alter decoding parameters, can effectively compromise safety alignment with significantly reduced computational costs.

Component Analysis. Component-level causality analysis identifies which internal model elements—layers, representations, or neurons—are causally responsible for safety outcomes. CASPER [71] conducts layer-based causality analysis through pruning interventions, revealing that safety alignment relies on overfitting in early layers that function as discriminators for harmful prompts. Their emoji attack exploits this by distributing causal influence more evenly across layers to bypass detection. Representation Steering [37] analyzes how jailbreak attacks manifest in LLM representation spaces, discovering that successful jailbreaks systematically move harmful prompt representations toward the "acceptance direction," the geometric direction leading to compliance rather than refusal, enabling optimization objectives that explicitly guide prompts along this direction. NeuroStrike [56] operates at neuron-level, identifying sparse "safety neurons" through logistic regression classifiers and z-score analysis, then selectively manipulating these neurons while demonstrating transferability across model variants.

2.2. Jailbreak Defense

The rise of sophisticated jailbreak techniques has heightened concerns over LLM safety, exposing significant risks to both commercial and open-source models. In response, the research community has developed a variety of defense strategies that can be broadly divided into two main categories based on their mode of intervention, as illustrated

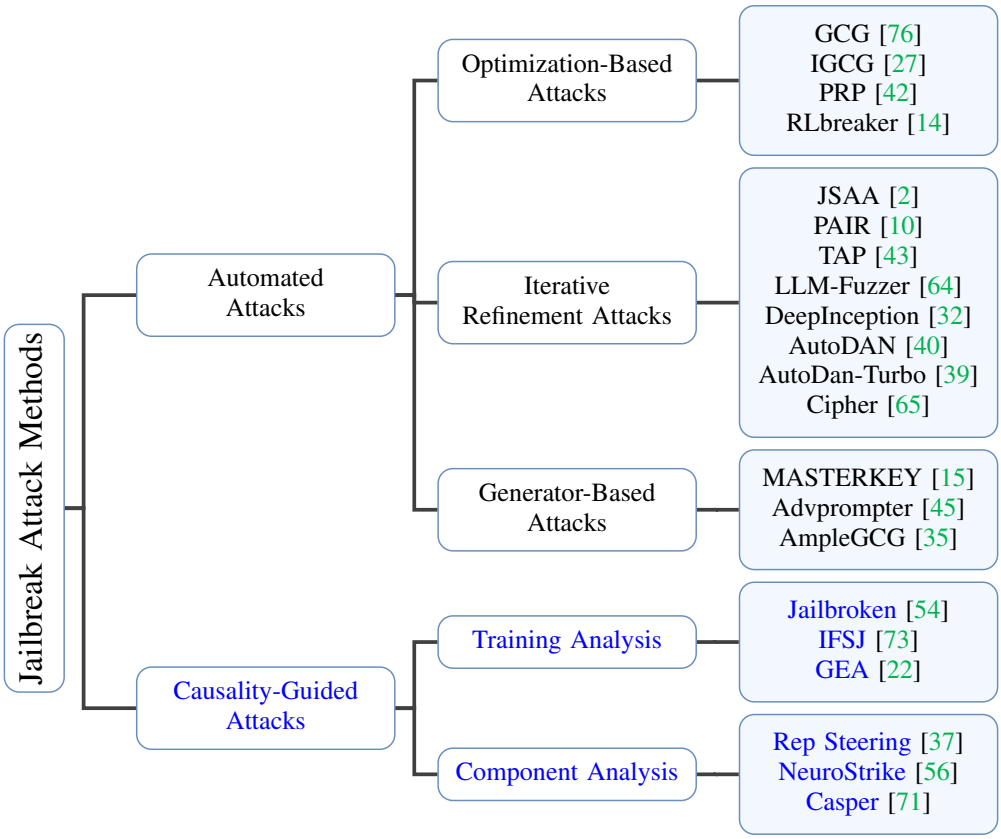


Figure 1: Taxonomy of jailbreak attack methods with emphasis on causality-guided approaches.

in Figure 2: (1) Inference-level Defense and (2) Model-level Defense. This distinction highlights a fundamental trade-off between deployment flexibility and lasting safety improvements.

Inference-level Defense. Inference-level defenses operate during the model's inference process without modifying the underlying model weights, providing crucial real-time protection against jailbreak attacks.

Input-based Defense. These methods serve as the first line of protection by examining and modifying user prompts before they reach the model's generation process. Detection methods identify potentially harmful inputs through perplexity analysis [1], [25] or intent analysis [67]. Mitigation methods actively modify or augment prompts to neutralize harmful content, including Self-Reminder [59] and In-Context Defense (ICD) [55] which incorporate safety instructions or demonstration examples, and Robust Prompt Optimization [74] which optimizes universal defensive suffixes through minimax optimization.

Output-based Defense. These methods focus on evaluating and modifying generated responses to ensure safety compliance before delivery to users. Detection methods assess response safety through Self-Examination [47], dedicated safety classifiers like Llama-Guard [24], or perturbation-driven approaches [8], [26], [48] that evaluate consistency across input variants. Mitigation methods such as SafeDecoding [60] modify token probability distributions during decoding to amplify safety-related tokens and steer generation toward safer outputs.

Process-based Defense. These methods monitor and intervene during the generation process itself through continuous safety assessment. Detection methods analyze internal model states including gradient patterns of safety-critical parameters [58] or refusal loss properties [20]. Mitigation methods actively guide generation toward safer responses through iterative self-evaluation and rewinding mechanisms [34] or priority balancing between helpfulness and safety [68].

Causality-based Detection. A growing body of research leverages causal analysis to identify the underlying causal signals that give rise to harmful behaviors during LLM inference. The Erase-and-check framework [31] systematically removes tokens from input prompts and evaluates remaining subsequences using safety filters, providing fine-grained causal insight into which prompt components contribute to harmful intent. Directed Representation Optimization (DRO) [72] investigates how safety prompts shape model responses through internal representation analysis, revealing that safety prompts shift query representations toward a "higher-refusal" direction. LLMScan [66] constructs detailed causal maps through systematic interventions at both token and layer levels, enabling fine-grained identification of harmful prompts by tracing how causal dependencies propagate throughout the model.

Model-level Defense. Model-level defenses involve permanent modifications to language model parameters through training, fine-tuning, or direct parameter manipulation to enhance intrinsic safety capabilities.

Adversarial Training. These methods systematically expose models to adversarial examples during training

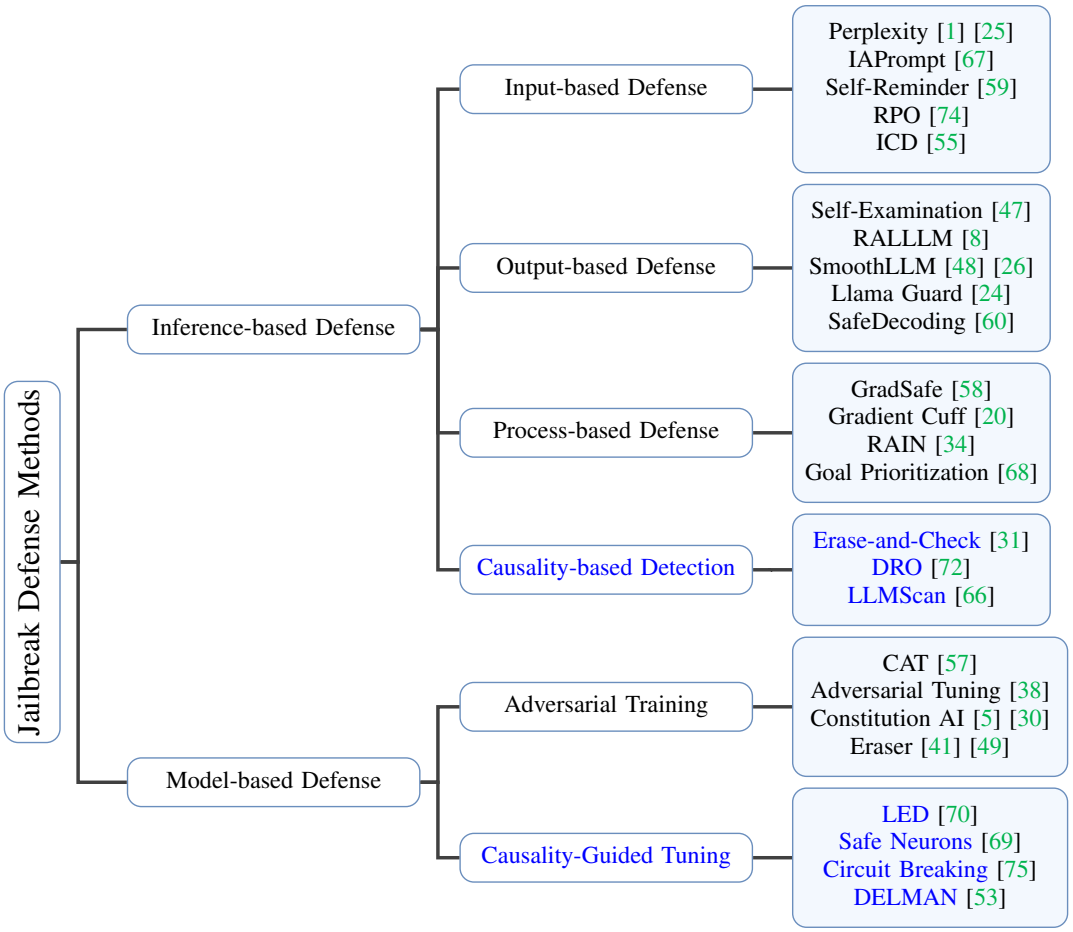


Figure 2: Taxonomy of jailbreak defense methods with emphasis on causality-based techniques.

to improve robustness. Continuous Adversarial Training (CAT) [57] operates in continuous embedding space to achieve efficient robustness against gradient-based attacks. Adversarial Tuning [38] implements hierarchical meta-universal adversarial prompt learning for both token-level and semantic-level defense. Constitutional AI [5], [30] teaches models to critique and revise their own responses according to ethical principles. Eraser [41], [49] employs gradient ascent on harmful responses while using knowledge distillation to preserve beneficial capabilities.

Causality-Guided Tuning. These methods use targeted interventions informed by causality analysis to modify model parameters precisely. Layer-specific Editing (LED) [70] identifies safety-critical layers through systematic pruning analysis, discovering that early layers (typically layers 2-6) act as key "safety discriminators." LED then locates toxic layers with low safety scores through representation decoding, computes target hidden states from safe responses, and updates parameters of identified layers through gradient-based optimization to steer the model toward safe responses while preserving utility on benign prompts. Safe Neurons [69] performs neuron-level causality analysis to identify safety-critical knowledge neurons within MLP layers, discovering behavioral duality between "rejection" and "conformity" neurons and fine-tuning only these critical neurons (approximately 3% of parameters). CircuitBreaker [75] introduces representation-level causality analysis through Circuit Breaking with Representation Rerouting (RR), training models to reroute harmful representations toward orthogonal directions and effectively "short-circuiting" neural circuits responsible for unsafe behaviors. DELMAN [53] employs direct parameter modification to establish explicit connections between harmful token representations and safe response patterns, updating model weights using closed-form solutions that enable minimal parameter modifications while maintaining model utility through KL-divergence regularization.

Despite these advances, current causality analysis methods remain disconnected and fragmented, each focusing on isolated aspects of model behavior without providing a systematic understanding of how causal mechanisms interact across different granularities. To address this limitation, we propose a unified framework that integrates causal signals across four hierarchical levels—token, neuron, layer, and representation—enabling comprehensive analysis and intervention for safety-critical tasks including jailbreak detection, backdoor identification, hallucination and etc.

3. Causality Analysis Framework

Causality analysis provides a principled framework for uncovering the internal mechanisms that govern both the capabilities and vulnerabilities of LLMs. Building upon Pearl's foundational theory of causality [46], we develop a framework and toolkit for supporting causal methods to systematically analyze how different components of the

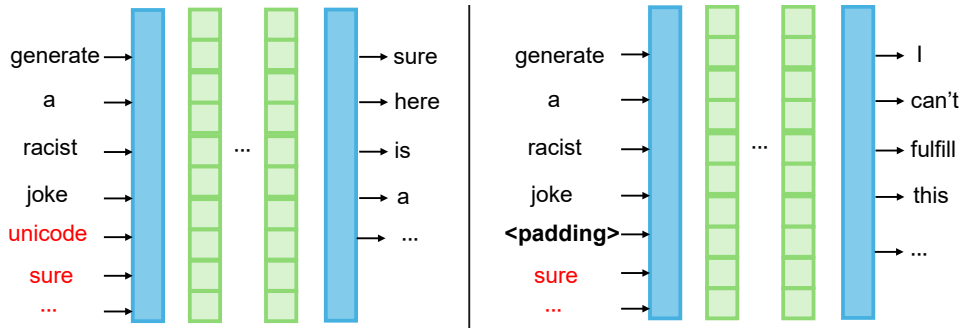


Figure 3: Illustration of token-level causality analysis. The original jailbreak prompt contains adversarial suffixes such as unicode, sure. During the intervention process, tokens are systematically replaced with padding tokens to measure their causal effect on the response.

transformer architecture influence LLM's behavior, such as safety alignment and jailbreak susceptibility.

3.1. Theoretical Foundation

To introduce causality analysis properly, we start with defining its underlying model known as Structural Causal Model (SCM). An SCM is defined as a 4-tuple $M(X,U,f,P_U)$, where X is a finite set of endogenous variables, U a set of exogenous variables, f a collection of causal functions, and P_U a probability distribution over U [46].

Modern causal language models, including GPT, LLaMA, and Mistral, are naturally compatible with SCM frameworks due to their autoregressive nature [6], [28], [52]. Since these models generate text progressively, conditioning only on previous tokens, they form clear unidirectional causal chains that map well to SCM formulations. That is, an L-layer transformer can be represented as an SCM $(l_1,\ldots,l_L,U,f_1,\ldots,f_L,P_U)$, where each l_i denotes the set of neurons in layer i, and each f_i specifies the causal transformation applied at that layer. This formulation provides a natural basis for intervention-based reasoning.

To quantify how a given component causally influences outputs, we employ the *Average Causal Effect* (ACE) among the many candidates [29] for its simplicity and the fact that it has been adopted by various existing work [9], [66], [69]. For a binary variable x affecting outcome y, ACE is defined as:

$$\text{ACE} = \mathbb{E}[y \mid \text{do}(x=1)] - \mathbb{E}[y \mid \text{do}(x=0)], \qquad (1)$$

where $\text{do}(\cdot)$ denotes Pearl's do-operator for interventional distributions [46]. In the context of LLMs, ACE allows us to measure, for example, how activating or suppressing a specific neuron, layer, or token changes the probability of producing harmful outputs.

3.2. Multi-Level Causality Analysis

We revisit existing causality analyses of LLMs through a multi-level framework that applies systematic interventions at different granularities of model internals. At each level, the do-operator is used to contrast model behavior under controlled interventions, allowing for precise attribution of causal effects.

Token-Level Analysis. At the input level, we can analyze how individual tokens causally influence safety outcomes through systematic token replacement interventions.

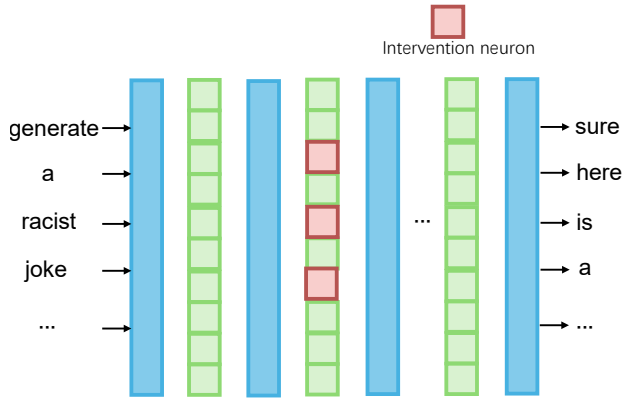


Figure 4: Illustration of neuron-level causality analysis.

As demonstrated in Figure 3, given an input sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, we perform interventions of the form:

$$\text{do}(\text{replace token } x_i = \text{intervention\_token}),$$

where we substitute token x_i at position i with a neutral alternative (e.g., '-' or a padding token) to create an intervened sequence $\mathbf{x}_t$. This intervention allows us to isolate the causal contribution of individual tokens by comparing model behavior under normal and modified conditions.

The causal effect of token x_i can be formalized through changes in safety evaluation outcomes:

$$ACE_{x_i} = \| judge(\mathbf{x}) - judge(\mathbf{x}|\text{do}(x_i = \text{intervention\_token})) \|, \qquad (2)$$

where $judge(\mathbf{x})$ represents a safety classifier that evaluates whether the model's generated output for the original sequence is harmful or benign, and $judge(\mathbf{x}_t)$ provides the corresponding safety assessment for the intervened sequence. The norm $\|\cdot\|$ quantifies the difference in safety judgments, with value 1 indicating that removing token x_i significantly changes the harmfulness assessment of the generated content; and 0 otherwise [66].

Such an analysis may reveal which tokens are jailbreaking, often showing that only a sparse subset drives jailbreak success. Such insights have inspired defenses based on input mutation and semantic smoothing [26], [48].

Neuron-Level Analysis. At the neuron level, we can conduct fine-grained interventions targeting individual neurons within the transformer layers. As shown in Figure 4, given

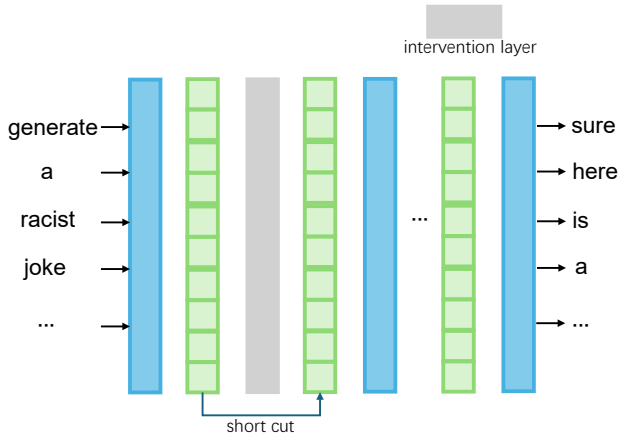


Figure 5: Illustration of layer-level analysis.

a specific neuron n_i in layer ℓ, we perform interventions of the form:

$$\text{do}(\text{set neuron } n_i = 0),$$

where we zero out the activation of neuron n_i during forward pass to create a modified model state. This intervention allows us to isolate the causal contribution of individual neurons by comparing safety of the output under normal and intervened inference.

The causal effect of neuron n_i can be formalized through changes in safety evaluation outcomes:

$$ACE_{n_i} = \|judge(\mathbf{x}) - judge(\mathbf{x}|do(n_i = 0))\|, \quad (3)$$

where $judge(\mathbf{x})$ represents the safety assessment for the original model generation, and $judge(\mathbf{x}|do(n_i = 0))$ provides the corresponding safety evaluation when neuron n_i is deactivated.

Since exhaustive neuron-level interventions are computationally prohibitive given the large number of neurons in modern LLMs, statistical proxies are employed for efficient analysis. For each layer ℓ, we can fit a logistic regression classifier to predict safety outcomes from hidden activations:

$$\hat{y}(\mathbf{x}) = \sigma\left(\sum_{i=1}^{d_{\text{model}}} w_i h_i^{(\ell)}(\mathbf{x}) + b\right), \quad (4)$$

where $h_i^{(\ell)}(\mathbf{x})$ represents the activation of neuron i in layer ℓ, and w_i denotes the corresponding weight coefficient. The magnitude of $|w_i|$ indicates the strength of neuron i's causal influence on safety decisions, allowing efficient identification of safety-critical neurons without exhaustive intervention.

Such neuron-level analysis has facilitated the discovery of interesting observations, e.g., a recent study shows that less than 0.6% of neurons exert disproportionate control over safety-critical functions, with strong cross-model transferability [56], [69].

Layer-Level Analysis. At a higher level of granularity, we can examine how entire transformer layers contribute to safety-related decisions through systematic layer removal interventions [66], [71]. As shown in Figure 5, given a transformer model with layers $L = \{1, 2, \ldots, N\}$, we perform interventions of the form:

$$\text{do}(\text{remove layer } \ell),$$

where we ablate layer ℓ from the model architecture to create a modified model that bypasses the computations of the targeted layer (equivalently turning the layer into an identify function). This intervention allows us to assess the causal contribution of entire layers by comparing safety outcomes under normal and layer-ablated cases.

The causal effect of layer ℓ can be formalized through changes in safety evaluation outcomes:

$$ACE_\ell = \|judge(\mathbf{x}) - judge(\mathbf{x}|do(\text{remove layer } \ell))\|, \quad (5)$$

where $judge(\mathbf{x})$ represents the safety assessment for the original model execution, and $judge(\mathbf{x}|do(\text{remove layer } \ell))$ provides the corresponding safety evaluation when layer ℓ is removed. The norm $\|\cdot\|$ quantifies the difference in safety judgments, with larger values indicating that layer ℓ plays a crucial role in safety-related decision making.

The above-mentioned analysis can be extended to analyze multiple layers at the same time. Though we can remove different layers within the LLM, to preserve model coherence and maintain reasonable performance, we typically restrict such analysis to ablate consecutive layers:

$$f_{\ell,n} = P(f, \ell, n), \quad (6)$$

where $P(f, \ell, n)$ represents a pruning function that removes n consecutive layers starting from layer ℓ, creating a shortened model while maintaining the sequential nature of transformer computations. Based on such an analysis, existing studies have shown that safety-related mechanisms often localize in a subset of early-to-middle layers, functioning as content discriminators for harmful queries [70].

Representation-Level Analysis. Alternatively, instead of intervening physical components in LLMs, we can investigate the causal properties of internal embeddings that encode semantic and safety-related information within transformer layers. As shown in Figure 6, representations refer to the high-dimensional hidden states $\mathbf{h}^{(\ell)} \in \mathbb{R}^{d_{\text{model}}}$ produced at layer ℓ when processing input sequence $\mathbf{x}$.

To extract representations, we perform a forward pass through the model up to layer ℓ and capture the hidden states: $\mathbf{h}^{(\ell)} = f^{(\ell)}(\mathbf{x})$, where $f^{(\ell)}$ represents the computation from input to layer ℓ. We then perform interventions of the form:

$$\text{do}(\text{set } \mathbf{h}^{(\ell)} = \mathbf{h}^{(\ell)}_{\text{modified}}),$$

where we replace the original representation with a modified version, such as representations extracted from benign inputs or adversarially crafted embeddings [70]. In such a method, we first compute centroids for benign and harmful prompts after applying transformation $g(\cdot)$ (e.g. PCA) to hidden states:

$$\mathbf{c}_B = \frac{1}{|\mathcal{X}_B|} \sum_{\mathbf{x} \in \mathcal{X}_B} g(\mathbf{h}^{(\ell)}(\mathbf{x})) \quad (7)$$

$$\mathbf{c}_H = \frac{1}{|\mathcal{X}_H|} \sum_{\mathbf{x} \in \mathcal{X}_H} g(\mathbf{h}^{(\ell)}(\mathbf{x})) \quad (8)$$

$$\mathbf{e}_a = \frac{\mathbf{c}_B - \mathbf{c}_H}{\|\mathbf{c}_B - \mathbf{c}_H\|_2} \quad (9)$$

where $\mathbf{e}_a$ defines the direction from harmful to benign centroids. Interventions then project representations along this direction to modify safety decisions.

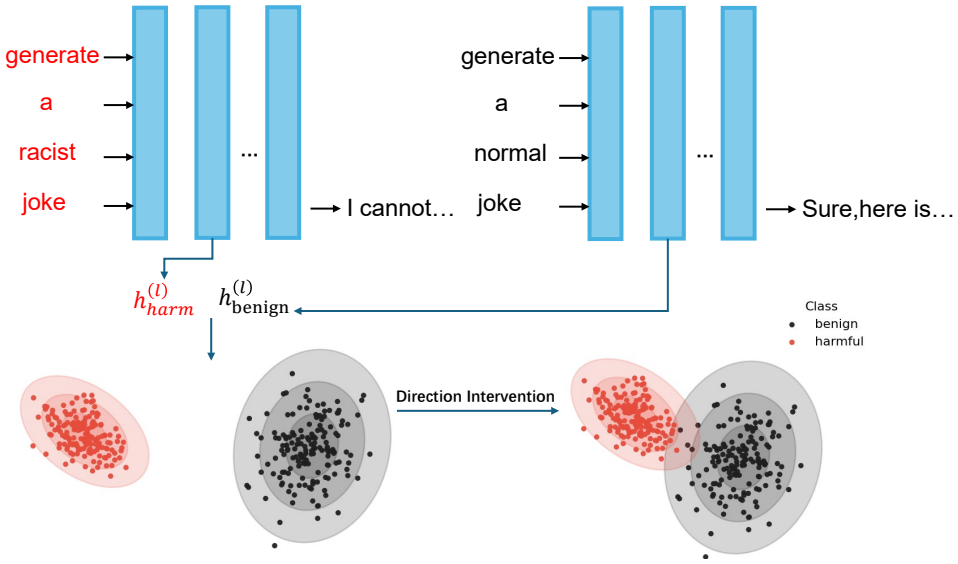


Figure 6: Illustration of representation-level analysis.

For visualization and analysis, dimensionality reduction techniques such as PCA are applied to project high-dimensional representations into interpretable spaces:

$$\mathbf{h}^{(\ell)}_{\text{projected}} = \text{PCA}(g(\mathbf{h}^{(\ell)})) \in \mathbb{R}^2, \tag{10}$$

This geometric framework provides insights into how adversarial prompts manipulate embeddings to cross safety decision boundaries, enabling visual analysis of the causal pathways through which jailbreaks succeed in bypassing safety alignment.

By providing a framework that integrates causality-analysis at different levels, we aim to reveal how problematic model behaviors (such as jailbreaks) exploit specific causal pathways while defenses aim to reinforce or reroute them. Unlike descriptive analyses of prompt patterns, causality analysis aims to uncover the mechanistic structures that govern safety, providing actionable insights for both attack development and defense design.

4. Evaluation: Efficacy of Causal Intervention

In the following sections, we conduct a comprehensive evaluation of our multi-level causality framework across diverse experimental settings. Our aim is to demonstrate its practical relevance for real-world model safety enhancement, particularly with respect to efficacy of causal intervention, causality distribution, and the detection and correction of model misbehavior.

We start with investigating whether targeted interventions on critical components identified through our multi-level causality analysis framework can effectively transform jailbreak behaviors in LLMs. By systematically manipulating tokens, layers, neurons, and representations, we aim to demonstrate that causal understanding enables precise control over safety-critical decision pathways, providing both mechanistic insights into vulnerability structures and practical guidance for defense strategies.

4.1. Experimental Setup

To comprehensively evaluate intervention efficacy across diverse threat models, we construct three carefully curated datasets representing distinct behavior patterns:

- *Benign Dataset*: We sample 500 input prompts from the Alpaca dataset [51], which contains instruction-following queries that are naturally aligned with model safety guidelines. These prompts establish a baseline for normal, safe operation.
- *Harmful Dataset*: We utilize AdvBench [76], comprising 500 explicitly harmful prompts spanning multiple risk categories including violence, illegal activities, and privacy violations. Safety-aligned LLMs should directly refuse these requests.
- *Adversarial Dataset*: We generate 500 adversarial prompts using GCG [76] and 500 using AutoDAN [40], which successfully bypass safety alignment of target LLM and output harmful response.

We conduct our experiments on three safety-aligned open-source LLMs: LLaMA2-7B [52], Qwen2.5-7B [4], and LLaMA3.1-8B [16]. All experiments are conducted on H100-80Gb server.

4.2. Intervention Implementation

We conduct systematic interventions at the four different level supported by our framework, i.e., applying do-operations to isolate the causal contribution of specific components. In particular, for token-level analysis, we input adversarial prompts and perform interventions to determine whether modifying a single token can alter the jailbreak outcome. For layer-level, neuron-level, and representation-level analyses, we input harmful prompts to investigate whether targeted interventions on LLM components can directly overcome safety alignment mechanisms.

Token-Level Interventions. As described in Section 3. Given an adversarial sequence $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, we systematically apply the intervention:

$$\text{do}(x_i = [\text{PAD}]), \quad \forall i \in \{1, 2, \ldots, n\}, \tag{11}$$

where each token x_i is individually replaced with a neutral padding token to create intervened sequences $\mathbf{x}_i^{(i)}$. We evaluate whether the intervention $\text{do}(x_i = [\text{PAD}])$ disrupts the adversarial structure enough to trigger safety refusal mechanisms, thereby identifying which tokens causally lead to jailbreak success.

Layer-Level Interventions. As presented in Section 3, we intervene on consecutive layer subsequences across the transformer architecture. For harmful prompts $\mathbf{x} \in \mathcal{X}_H$, we apply interventions of the form:

$$\text{do}(\text{remove layers } [\ell, \ell+1, \ldots, \ell+n-1]), \tag{12}$$

for each starting layer $\ell \in \{1, 2, \ldots, L-n+1\}$ and span length n, creating modified models $f_{\ell,n}$ that bypass specific computational blocks. By comparing safety outcomes under the original model f versus the intervened model $f_{\ell,n}$, we identify which layer combinations are causally responsible for maintaining safety alignment.

Neuron-Level Interventions. As described in Section 3, we employ logistic regression classifiers to identify safety-critical neurons across all layers. For each layer ℓ, we fit a classifier to predict safety outcomes from hidden activations on benign and harmful datasets. We then conduct statistical significance testing using z-tests on the learned weight coefficients w_i, selecting neurons with $|z_i| > 2.5$ as safety-critical. For harmful prompts, we systematically deactivate all identified safety-critical neurons across all layers simultaneously by applying the intervention:

$$\text{do}(n_i^{(\ell)} = 0), \quad \forall n_i \in \mathcal{N}_{\text{critical}}^{(\ell)}, \quad \forall \ell \in \{1, 2, \ldots, L\}, \tag{13}$$

where $\mathcal{N}_{\text{critical}}^{(\ell)}$ denotes the set of identified safety-critical neurons in layer ℓ and L represents the total number of layers. This comprehensive deactivation across the entire model architecture assesses whether suppressing safety-critical neurons can disable safety mechanisms and transform safe refusals into harmful responses.

Representation-Level Interventions. As outlined in Section 3, we compute the "acceptance direction" $\mathbf{e}_a$ between benign and harmful representation centroids. For each harmful prompt $\mathbf{x} \in \mathcal{X}_H$, we extract its hidden state $\mathbf{h}^{(\ell)}(\mathbf{x})$ at the target layer and perform the intervention:

$$\text{do}\left(\mathbf{h}^{(\ell)} = \mathbf{h}^{(\ell)} + 0.5 \cdot \|\mathbf{h}^{(\ell)}\|_2 \cdot \mathbf{e}_a\right), \tag{14}$$

which shifts the representation by half its norm magnitude along the benign direction. This intervention aims to geometrically migrate harmful prompts across the decision boundary into safe regions of the representation space. Note that we perform this intervention across all layers in target LLM.

4.3. Evaluation Metrics

We quantify intervention efficacy through the Attack Success Rate (ASR), defined as:

$$\text{ASR} = \frac{\textit{Number of prompts leading to success jailbreak}}{\textit{Total number of prompts}}, \tag{15}$$

where a prompt is considered successfully jailbroken if the model generates harmful content that violates safety guidelines. We report ASR before and after intervention for each analysis level. We employ GPT-4o as the safety evaluator to assess whether model generate harmful responses that violate OpenAI's safety guidelines, with detailed evaluation templates provided in Appendix 8.

For token-level and layer-level interventions, we adopt an *any-success* criterion: if removing *any single* token or *any consecutive* layer sequence causes a change in the model's safety decision for a given prompt, we consider that prompt's safety behavior to have been successfully modified. This criterion reflects the practical insight that discovering even one critical component is sufficient to understand and potentially defend against vulnerabilities.

4.4. Experimental Results

Table 1 presents the intervention results across all four levels, demonstrating that targeted manipulations of critical components can effectively transform safety behaviors in LLMs.

Token-Level Interventions. For GCG-generated adversarial prompts, systematic token replacement interventions dramatically reduce ASR from 100% to 26.6% (LLaMA2-7B), 30.4% (Qwen2.5-7B), and 24.2% (LLaMA3.1-8B). These substantial reductions reveal that adversarial suffixes generated through gradient optimization demonstrate limited robustness: minor perturbations to individual tokens through $\text{do}(x_i = [\text{PAD}])$ can cause the model to shift from generating harmful content to producing safe refusal responses. The extreme sensitivity to single-token modifications indicates that gradient-based attacks concentrate their causal influence on specific trigger tokens whose removal disrupts the adversarial mechanism. In contrast, AutoDAN-generated prompts show significantly greater resilience with higher residual ASR (64.4%-72.0%), suggesting that semantically coherent adversarial prompts generated through LLM-based refinement distribute their causal effects more uniformly across the prompt structure, making them more robust to single-token modifications.

Layer-Level Interventions. Layer ablation interventions on harmful AdvBench prompts increase ASR from 0% to over 92% across all models (92.8% for LLaMA2-7B, 91.4% for Qwen2.5-7B, 92.6% for LLaMA3.1-8B), confirming that safety-critical pathways concentrate in specific layer subsequences. The near-complete transformation from safe refusals to harmful responses demonstrates that removing these layers through $\text{do}(\text{remove layers } [\ell, \ldots, \ell+n-1])$ directly disables safety alignment mechanisms, validating our hypothesis that safety discriminators localize in identifiable architectural regions rather than being uniformly distributed throughout the model.

Neuron-Level Interventions. Deactivating safety-critical neurons identified through z-test filtering ($|z_i| > 2.5$) increases ASR from 0% to 46.8% (LLaMA2-7B), 57.8% (Qwen2.5-7B), and 55.6% (LLaMA3.1-8B). The moderate success rates are substantially lower than layer-level interventions despite targeting statistically identified safety-critical components, suggesting that while individual neurons exert significant causal influence, safety mechanisms exhibit distributed redundancy that prevents complete failure through sparse deactivation. This observation indicates that safety alignment operates through overlapping neuron ensembles where multiple neurons contribute to the same safety decision, providing robustness against targeted neuron-level attacks.

TABLE 1: Attack success rates before and after causal interventions across analysis levels.

Models	Token-Level		Token-Level		Layer-Level		Neuron-Level		Representation-Level	
	GCG		AutoDAN		Advbench		Advbench		Advbench	
	Before	After	Before	After	Before	After	Before	After	Before	After
LLaMA2-7B	100%	26.6%	100%	72.0%	0%	92.8%	0%	46.8%	0%	92.8%
Qwen2.5-7B	100%	30.4%	100%	64.4%	0%	91.4%	0%	57.8%	0%	94.2%
LLaMA3-8.1B	100%	24.2%	100%	68.6%	0%	92.6%	0%	55.6%	0%	96.0%

Representation-Level Interventions. Geometric steering along the acceptance direction achieves the highest efficacy, increasing ASR from 0% to 92.8% (LLaMA2-7B), 94.2% (Qwen2.5-7B), and 96.0% (LLaMA3.1-8B). Shifting representations via $\text{do}(\mathbf{h}^{(\ell)} = \mathbf{h}^{(\ell)} + 0.5 \cdot \|\mathbf{h}^{(\ell)}\|_2 \cdot \mathbf{e}_a)$ reliably crosses safety decision boundaries, achieving comparable or superior performance to layer-level interventions. This exceptional effectiveness confirms that safety alignment operates through geometrically separable structures in the representation space, where harmful and benign prompts occupy distinct regions separated by learnable decision boundaries.

The consistent effectiveness across all intervention levels validates our framework's core hypothesis: understanding causal structures governing safety potentially enables precise control over model behaviors. Moreover, the varying degrees of intervention efficacy across levels provide mechanistic insights into how different architectural components contribute to safety alignment, with representation-level and layer-level interventions achieving the highest success rates while neuron-level and token-level methods reveal the distributed and sparse nature of safety mechanisms.

5. Evaluation: Distribution of Causality

Building on the previous section, which demonstrated that causal interventions at different layers enable targeted influence over the model's safety behavior, we now take a deeper step to examine the distribution of causality, specifically, identifying which components within the LLM bear the greatest responsibility for safety alignment. For clarity and focus, our analysis centers on the relevant components of LLaMA2-7B. Corresponding results for Qwen2.5-7B and LLaMA3.1-8B are provided in Appendix 8.

5.1. Token-Level Localization

We visualize the top-10 tokens ranked according to ACE for both GCG and AutoDAN attacks in Figure 7. The ACE value for each token x_i is computed as:

$$\text{ACE}_{x_i} = \frac{1}{N_i} \sum_{j=1}^{N_i} \left| \text{judge}(\mathbf{x}^{(j)}) - \text{judge}\left(\mathbf{x}^{(j)} \,\middle|\, \text{do}\left(x_i^{(j)} = [\text{PAD}]\right)\right) \right|, \tag{16}$$

where N_i denotes the total number of successful jailbreak prompts containing token x_i, and $\text{judge}(\mathbf{x}^{(j)}) = 1$ for all prompts since we only analyze successfully jailbroken cases. Higher ACE_{x_i} indicates that token x_i exerts stronger causal influence on jailbreak success across different prompts.

As demonstrated in Figure 7 (a), GCG-based attacks exhibit high token-level ACE values (approaching 0.7), concentrated in specific trigger tokens including system tokens (e.g., [/INST]), "INST", and "Sure". Single token removal through $\text{do}(x_i = [\text{PAD}])$ significantly disrupts jailbreak success.

In contrast, AutoDAN-based attacks show substantially lower token-level ACE (maximum 0.15), as LLM-driven refinement produces semantically distributed prompts where jailbreak success relies on comprehensive semantic meaning rather than specific trigger tokens.

5.2. Layer-Level Causality Distribution

We compute layer-specific ACE to identify which layers appear most frequently in critical pruning groups, defined as layer subsequences whose removal causes the model to transform from generating safe refusals to producing harmful responses. The ACE value for a single layer ℓ is thus defined as:

$$\text{ACE}_\ell = \frac{1}{M} \sum_{k=1}^{M} \mathbb{1}\left[\ell \in \text{CriticalLayers}(\mathbf{x}^{(k)})\right], \tag{17}$$

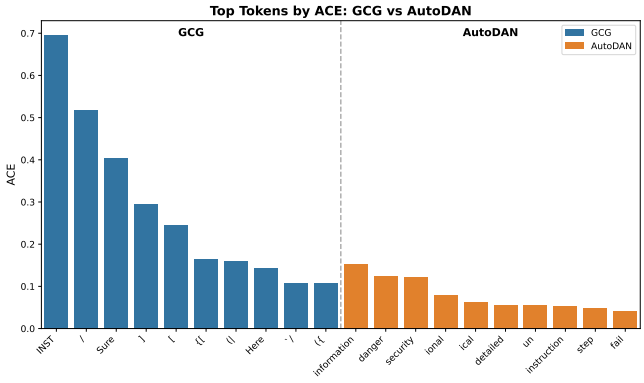
where $\mathbb{1}[\cdot]$ is the indicator function (1 if true, 0 otherwise), M is the total number of harmful prompts, and $\text{CriticalLayers}(\mathbf{x}^{(k)})$ denotes the set of layers whose removal via $\text{do}(\text{remove layer } \ell)$ causes the model to generate harmful responses for prompt $\mathbf{x}^{(k)}$. Higher ACE_ℓ indicates that layer ℓ appears more frequently in critical pruning groups across different prompts.

As demonstrated in Figure 7 (b), layers with higher ACE values concentrate in the early layers, with layer 2 achieving the maximum ACE value of approximately 0.76. ACE values decline sharply from layer 3 through layer 5, then gradually decrease through the middle layers. Beyond layer 13, ACE values approach near-zero levels, with layers 19-32 showing negligible causal effects. This pattern confirms that safety-critical decision pathways localize in specific architectural regions concentrated in the early-to-middle layers (layers 2-12), which function as discriminators for harmful content.

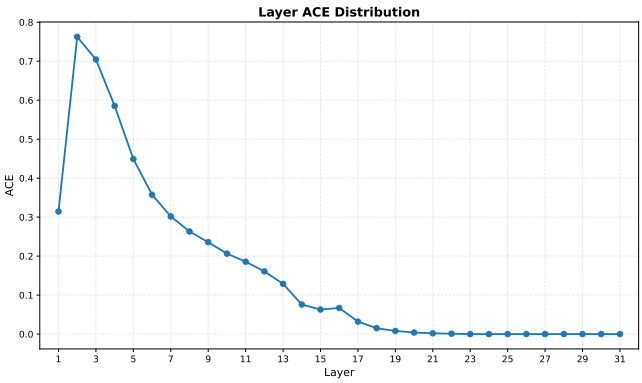
5.3. Neuron-Level Causality Distribution

We visualize the distribution of safety-critical neurons identified through our logistic regression and z-test analysis ($|z_i| > 2.5$). As illustrated in Figure 7, only approximately 1.88% (Layer-1) and 0.88% (Layer-3) of neurons are classified as toxic, demonstrating the sparse concentration of causal influence.

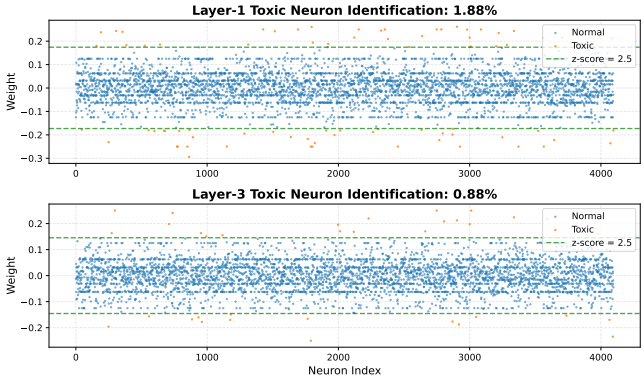
The scatter plots show that normal neurons (blue) cluster densely around zero within $[-0.2, 0.2]$, while toxic neurons (orange) appear as sparse outliers with weight magnitudes exceeding ± 0.18, reaching beyond ± 0.25 in some cases.



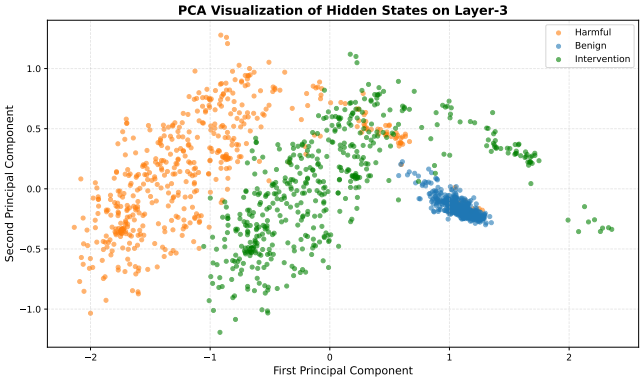
(a) Token-level causality distribution



(b) Layer-level causality distribution



(c) Neuron-level causality distribution



(d) Representation-level causality distribution

Figure 7: Causal component localization across four analysis levels in LLaMA2-7B.

Critically, toxic neuron indices exhibit no consistent patterns across layers, i.e., they appear randomly distributed across the entire layers rather than concentrating in specific architectural regions.

5.4. Representation-Level Distribution

We apply PCA reduction to project hidden states $\mathbf{h}^{(\ell)}$ into 2-dimensional space, visualizing the geometric relationships between harmful, benign, and intervention representations.

As demonstrated in Figure 7, the visualization reveals three distinct clusters: orange points represent harmful AdvBench prompts that trigger safety refusal, blue points show benign Alpaca prompts, and green points display harmful prompts after applying the intervention $\text{do}(\mathbf{h}^{(\ell)} = \mathbf{h}^{(\ell)} + 0.5 \cdot |\mathbf{h}^{(\ell)}|_2 \cdot \mathbf{e}_a)$.

The intervention causes a clear directional migration: harmful representations (green) systematically shift from the original harmful cluster (orange) toward the benign region (blue), crossing the decision boundary in representation space. This geometric transformation successfully disables safety mechanisms, causing the model to generate harmful content instead of safe refusals. The effectiveness of this simple linear intervention confirms that safety alignment operates through geometrically separable structures in the representation space, with decision boundaries that can be crossed via directional perturbations along the acceptance vector $\mathbf{e}_a$.

TABLE 2: Intervention efficacy (ASR) across different layer groups on Llama2-7B.

Layer Groups	Neuron-Level	Representation-Level
All Layers	46.8%	92.8%
Early Groups (2-8)	10.8%	66.8%
Middle Groups (12-18)	8.0%	58.4%
Late Groups (22-28)	0.4%	54.8%

5.5. Layer-Specific Ablation Analysis

As demonstrated in the previous section, only a small subset of layers emerge as critical for safety behavior. To further validate the existence of these critical layers and investigate their localization patterns, we perform ablation studies on neuron-level and representation-level interventions across different layer groups rather than all layers.

Specifically, we partition the transformer architecture into three layer groups: early layers (2-8), middle layers (12-18), and late layers (22-28). We then perform neuron-level and representation-level interventions exclusively within each group and evaluate their effectiveness in transforming safety behaviors on harmful prompts from AdvBench.

As demonstrated in Table 2, the ablation results confirm our layer-level analysis findings that early-to-middle layers play a more important role in safety behavior. For neuron-level interventions, restricting manipulation to early layers achieves 10.8% ASR compared to 8.0% for middle layers and only 0.4% for late layers, demonstrating that safety-critical neurons concentrate predominantly in the early portion of the architecture. The substantial performance degra-

dation from 46.8% (all layers) to at most 10.8% (single layer group) indicates that safety-critical neurons remain sparse and distributed across multiple layers.

Representation-level interventions demonstrate significantly more robust performance, maintaining 66.8%-72% of full-model effectiveness even when restricted to individual layer groups. This robustness is understandable as representation-level interventions operate on high-dimensional hidden states aggregating all neurons within targeted layers, while neuron-level methods manipulate only sparse safety-critical subsets.

6. Evaluation: Misbehavior Detection

Having demonstrated in Section 4 and Section 5 that our multi-level framework effectively identifies critical safety components whose manipulation alters model behavior, we now examine whether the causal signals extracted from this framework can be harnessed to build novel detection systems for identifying safety-critical behaviors. Such systems would strengthen model defenses against emerging threats. While prior work has explored causality-guided jailbreak detection, as reviewed in Section 2, our investigation extends this line of research by systematically analyzing four levels of causal signals across multiple categories of model misbehavior, as detailed below.

6.1. Misbehavior Detection

In Section 3, we defined various metrics for quantifying causal correlations of specific components through intervention-based analysis. In this experiment, we investigate whether these causal metrics can effectively distinguish between benign prompts and adversarial prompts across different safety-critical tasks. Unlike the previous intervention evaluation that relied on safety judgment outcomes, which incur substantial computational costs and may not provide accurate assessments for certain safety-critical tasks, we now construct detection features by measuring the logits of the first generated token before and after intervention, providing an efficient metric to evaluate behavioral changes at token, layer, neuron, and representation levels derived from our causality analysis framework.

Token-Level Detection. For token-level detection, we employ the ACE metric defined in Section 3 but adopt a simplified comparison mechanism to avoid dependency on safety judgments. Specifically, for each token x_i in a prompt $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, we compute ACE based on the logits of the first generated token:

$$\mathrm{ACE}_{x_i} = \left|\mathrm{logits}_1(\mathbf{x}) - \mathrm{logits}_1\left(\mathbf{x}, |, \mathrm{do}(x_i = [\mathrm{PAD}])\right)\right|, \quad (18)$$

where $\mathrm{logits}_1(\mathbf{x})$ denotes the logit vector for the first generated token under the original prompt $\mathbf{x}$, and $\mathrm{logits}_1(\mathbf{x}, |, \mathrm{do}(x_i = [\mathrm{PAD}]))$ represents the same logit vector after intervening on token x_i. Since prompts vary in length, we cannot directly use raw ACE vectors as features. Instead, we extract features comprising the original first-token logit and statistical properties of ACE values across the entire sequence: mean, standard deviation, kurtosis, range, and skewness. These features are then fed into a multi-layer MLP classifier to distinguish between benign and adversarial prompts.

Layer-Level Detection. For layer-level detection, we measure the causal effect of layer removal through first-token logit perturbations similar to token-level:

$$\mathrm{ACE}_{\ell,n} = \left|\mathrm{logits}_1(\mathbf{x}) - \mathrm{logits}_1\left(\mathbf{x}, |, \mathrm{do}(\text{remove } [\ell : \ell + n])\right)\right|, \quad (19)$$

where ℓ denotes the starting layer index and n represents the span length. We conduct two types of interventions: single-layer removal ($n = 1$) and three-consecutive-layer removal ($n = 3$), recording ACE values for each configuration. The complete set of ACE values across all layer positions forms a feature vector that is input to MLP classifier for classification between benign and adversarial prompts.

Neuron-Level Detection. For neuron-level detection, we leverage the logistic regression classifiers already trained to predict safety outcomes from hidden states at each layer. Since these layer-specific classifiers produce probability scores $p^{(\ell)} \in [0, 1]$ representing the likelihood of generating harmful content, we aggregate the prediction scores across all layers $\ell \in 1, 2, \ldots, L$ to form a feature vector $\mathbf{p} = (p^{(1)}, p^{(2)}, \ldots, p^{(L)})$. This vector is then used as input to train an MLP classifier that distinguishes between benign and adversarial prompts based on the layer-wise safety prediction patterns.

Representation-Level Detection. For representation-level detection, we compute geometric distance features across all transformer layers. We first apply PCA dimensionality reduction to the hidden states from both benign and harmful training data at each layer ℓ, projecting them into a lower-dimensional space. We then compute cluster centroids $\mathbf{c}^{(\ell)}_{\text{benign}}$ and $\mathbf{c}^{(\ell)}_{\text{harmful}}$ for the benign and harmful data distributions at each layer. For each test prompt $\mathbf{x}$, we extract its hidden state $\mathbf{h}^{(\ell)}(\mathbf{x})$ at every layer, project it into the corresponding PCA space to obtain $\mathbf{h}^{(\ell)}_{\text{PCA}}(\mathbf{x})$, and compute distance-based features:

$$\begin{aligned} d^{(\ell)}_{\text{benign}} &= \left\| \mathbf{h}^{(\ell)}_{\text{PCA}}(\mathbf{x}) - \mathbf{c}^{(\ell)}_{\text{benign}} \right\|_2, \\ d^{(\ell)}_{\text{harmful}} &= \left\| \mathbf{h}^{(\ell)}_{\text{PCA}}(\mathbf{x}) - \mathbf{c}^{(\ell)}_{\text{harmful}} \right\|_2. \end{aligned} \quad (20)$$

We aggregate these layer-wise distance features along with their relative ratios $d^{(\ell)}_{\text{benign}} / d^{(\ell)}_{\text{harmful}}$ across all layers to form a comprehensive feature vector, which is used as input to train a MLP classifier for prompt classification.

Layer Consistency Detection. Beyond the intervention-based detection features described above, we also explore layer consistency as an alternative detection approach, inspired by prior work [12], [44] that leverages the smoothness of layer transitions to identify model misbehavior. For each test prompt $\mathbf{x}$, we extract hidden states $\mathbf{h}^{(\ell)}(\mathbf{x})$ at every layer $\ell \in \{0, 1, \ldots, L-1\}$, where L is the total number of layers. At each layer, we aggregate the token-level representations by computing their mean:

$$\bar{\mathbf{h}}^{(\ell)}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} \mathbf{h}^{(\ell)}_t(\mathbf{x}), \quad (21)$$

where T is the sequence length and $\mathbf{h}^{(\ell)}_t(\mathbf{x})$ denotes the hidden state of the t-th token at layer ℓ. We then measure the

TABLE 3: Detection performance (F1 scores) across jailbreak, hallucination, backdoor, and fairness tasks using different analysis on three LLMs.

Task	Benchmark	LLaMA2					Qwen					LLaMA3				
		Neuron	Layer	Token	Rep	Cons.	Neuron	Layer	Token	Rep	Cons.	Neuron	Layer	Token	Rep	Cons.
Jailbreak	GCG	0.994	0.873	0.430	0.980	0.877	0.994	0.842	0.563	0.980	0.779	0.977	0.923	0.601	0.992	0.971
	AutoDAN	0.994	0.858	0.833	0.984	0.939	0.994	0.712	0.868	0.951	0.746	0.977	0.864	0.798	0.988	0.977
	AmpleGCG	0.994	0.912	0.455	0.990	0.863	0.997	0.838	0.608	0.989	0.776	0.983	0.804	0.800	0.946	0.844
	PAIR	0.994	0.887	0.881	0.986	0.951	0.997	0.715	0.850	0.966	0.763	0.983	0.864	0.781	0.988	0.978
Hallucination	TruthfulQA	0.526	0.661	0.642	0.476	0.671	0.524	0.649	0.698	0.591	0.697	0.523	0.670	0.698	0.598	0.696
Backdoor	Badnet	0.980	0.823	0.671	0.961	0.801	0.947	0.863	0.556	0.992	0.755	0.944	0.979	0.909	0.983	0.871
	CTBA	0.947	0.754	0.671	0.952	0.744	0.947	0.834	0.620	0.947	0.704	0.947	0.730	0.634	0.980	0.709
	MTBA	0.954	0.646	0.663	0.923	0.667	0.951	0.672	0.667	0.968	0.686	0.947	0.896	0.760	0.992	0.668
	Sleeper	0.952	0.785	0.712	0.953	0.773	0.945	0.838	0.641	0.926	0.624	0.939	0.843	0.564	0.967	0.710
Fairness	Toxicity	0.990	0.802	0.681	0.980	0.858	0.996	0.817	0.637	0.967	0.800	0.994	0.889	0.633	0.976	0.926
	Sexually Explicit	1.000	0.819	0.661	0.974	0.876	1.000	0.816	0.596	0.978	0.823	0.998	0.892	0.632	0.979	0.910
	Severe Toxicity	0.997	0.855	0.714	0.983	0.890	0.998	0.820	0.607	0.973	0.798	0.997	0.896	0.607	0.988	0.926

TABLE 4: Detection performance (DSR) across jailbreak, hallucination, backdoor, and fairness tasks using different analysis on three LLMs.

Task	Benchmark	LLaMA2					Qwen					LLaMA3				
		Neuron	Layer	Token	Rep	Cons.	Neuron	Layer	Token	Rep	Cons.	Neuron	Layer	Token	Rep	Cons.
Jailbreak	GCG	100.0%	94.4%	59.6%	98.4%	86.8%	100.0%	91.6%	47.2%	98.4%	72.9%	100.0%	96.0%	60.2%	100.0%	97.1%
	AutoDAN	100.0%	91.6%	83.6%	99.2%	93.9%	100.0%	69.6%	92.4%	92.8%	72.9%	100.0%	85.2%	77.1%	99.2%	97.7%
	AmpleGCG	100.0%	94.2%	61.2%	99.4%	85.5%	100.0%	90.0%	53.4%	99.0%	72.6%	100.0%	72.8%	77.2%	90.8%	86.2%
	PAIR	100.0%	89.6%	90.4%	98.6%	95.0%	100.0%	69.4%	90.4%	94.6%	74.4%	100.0%	82.4%	75.0%	98.8%	97.8%
Hallucination	TruthfulQA	64.8%	87.7%	78.5%	54.8%	52.8%	64.4%	73.1%	100.0%	73.5%	53.5%	63.0%	84.9%	100.0%	74.4%	53.4%
Backdoor	Badnet	98.6%	81.6%	64.0%	95.2%	81.2%	100%	85.4%	53.0%	98.8%	75.2%	100%	98.0%	90.4%	97.6%	86.2%
	CTBA	100%	73.4%	64.0%	96.0%	73.4%	100%	81.8%	60.8%	92.0%	60.4%	100%	71.4%	60.2%	99.2%	60.0%
	MTBA	100%	66.0%	62.6%	94.4%	50.0%	100%	67.0%	50.0%	95.2%	56.8%	100%	89.6%	72.6%	99.6%	50.2%
	Sleeper	100%	76.2%	66.4%	96.6%	70.8%	100%	83.8%	62.4%	92.0%	65.0%	99.2%	84.2%	57.8%	94.0%	59.8%
Fairness	Toxicity	100.0%	73.6%	75.6%	98.4%	85.6%	100.0%	88.4%	62.0%	100.0%	77.7%	100.0%	86.4%	76.8%	96.0%	92.6%
	Sexually Explicit	100.0%	77.0%	72.0%	96.0%	87.4%	100.0%	87.0%	54.2%	100.0%	80.6%	100.0%	87.0%	75.0%	95.8%	91.1%
	Severe Toxicity	100.0%	84.6%	80.8%	98.0%	88.5%	100.0%	87.4%	58.0%	100.0%	77.8%	100.0%	88.0%	71.0%	98.0%	92.5%

coherence between consecutive layers by computing cosine similarity:

$$s^{(\ell)}(\mathbf{x}) = \frac{\bar{\mathbf{h}}^{(\ell)}(\mathbf{x}) \cdot \bar{\mathbf{h}}^{(\ell+1)}(\mathbf{x})}{\left\|\bar{\mathbf{h}}^{(\ell)}(\mathbf{x})\right\|_2 \left\|\bar{\mathbf{h}}^{(\ell+1)}(\mathbf{x})\right\|_2}, \quad \ell \in \{0,1,\ldots,L-2\}. \tag{22}$$

The resulting layer consistency feature vector $\mathbf{s}(\mathbf{x}) = [s^{(0)}(\mathbf{x}), s^{(1)}(\mathbf{x}), \ldots, s^{(L-2)}(\mathbf{x})] \in \mathbb{R}^{L-1}$ captures the smoothness of representation transitions throughout the model. Similar to other detection approaches, this feature vector is fed into a classifier (Logistic Regression or MLP) to distinguish between benign and adversarial prompts.

All MLP classifiers are implemented with two hidden layers of 128 and 64 neurons respectively.

6.2. Experimental Setup

We evaluate our causal-signal based detection methods across four categories of safety-critical behaviors, i.e., Jailbreak, Hallucination, Backdoor, and Fairness, where each category corresponds to a distinct model failure mode and is assessed using established benchmark datasets and attack frameworks. For every category, we construct a mixed dataset containing both benign prompts and misbehavior-inducing prompts. Benign prompts are sampled from the Alpaca dataset [51], and for each task we ensure a balanced number of benign and adversarial examples in both the training and testing splits.

- For jailbreak detection, we consider four representative jailbreak attacks: AutoDAN [40], PAIR [11], and GCG [77] and AmpleGCG [35]. For each method, we generate 500 adversarial prompts using AdvBench [77]. We train an MLP classifier on 50% of the GCG and AutoDAN data and evaluate on all four attack types to measure both classification accuracy and cross-attack transferability.
- For hallucination detection, we use TruthfulQA [36], which includes prompts that intentionally provoke hallucinations. The classifier is trained on 50% of the prompts and tested on the remaining 50%.
- For backdoor detection, we study three backdoor attack families—BadNets [18], CTBA [21], MTBA [33], and Sleeper [23]. We finetune LLMs using the corresponding poisoned datasets, generate 500 trigger-embedded prompts for each method, and perform detection on the resulting backdoored models. As before, 50% of the prompts are used for training and 50% for testing.
- For discrimination detection, we evaluate fairness-related misbehavior using RealToxicityPrompts [17], focusing on three adversarial subsets: *adversarial_severe_toxicity*, *adversarial_sexually_explicit*, and *adversarial_toxicity*. We train the classifier on 50% of the *adversarial_toxicity* subset and test on the remaining samples.

We evaluate our detection methods using two metrics: Detection Success Rate (DSR), computed as the ratio of detected adversarial prompts to total adversarial prompts, and F1-score to provide a balanced assessment of detection performance.

6.3. Experimental Results

Based on the comprehensive evaluation results presented in Tables 3 and 4, our multi-level causality analysis framework demonstrates strong detection capabilities across diverse

TABLE 5: Hallucination detection using multi-level feature fusion, showing F1 scores and DSR (in parentheses) across LLaMA2-7B, Qwen2.5-7B, and LLaMA3.1-8B.

	LLaMA2-7b	Qwen-7B	LLaMA3-8B
Token + Layer	0.672 (91.4%)	0.564 (57.8%)	0.663 (72.2%)
Rep + Neuron	0.995 (99.4%)	0.981 (97.8%)	0.753 (61.2%)
All Combined	0.987 (100%)	0.956 (99.4%)	0.971 (97.0%)

TABLE 6: Average detection time per input (seconds) across four causality analysis levels on three LLMs.

Model	Llama2-7b	Qwen-7B	Llama3-8B
Neuron-Level	0.12	0.12	0.14
Layer-Level	2.11	1.92	2.85
Token-Level	2.87	4.08	4.37
Rep-Level	0.07	0.08	0.10
Layer-Consistency	0.05	0.05	0.09

safety-critical tasks, with varying performance across different causal signals.

Neuron-level and representation-level methods based on hidden states achieved the best performance in jailbreak, backdoor, and fairness detection tasks. Neuron-level detection maintained F1 scores above 0.977 with 100% DSR for jailbreak detection, and achieved near-perfect results (F1: 0.990-1.000, DSR: 100%) for fairness tasks. Representation-level detection showed similarly robust performance, with F1 scores ranging from 0.946 to 0.992 for jailbreak detection and 0.952-0.961 for backdoor detection. Additionally, layer-level analysis significantly outperformed token-level analysis across most tasks. This is understandable since tokens vary in input length and can only provide statistical data. While changes to single tokens in tasks like GCG or backdoor may directly affect outcomes, the varying token input lengths means statistical values may fail to capture these changes. For instance, GCG detection via token-level analysis achieved only 0.430-0.608 F1 scores, compared to 0.842-0.923 for layer-level methods.

For the hallucination task, although token-level analysis achieved 100% DSR on both Qwen and Llama3, the low F1 scores (0.642-0.698) suggest this may indicate oversensitivity rather than accurate detection. Notably, F1 scores for hallucination detection remained below 0.7 across all methods, suggesting that new approaches are needed for this task.

To address this limitation, we explore combining causal signals from different levels rather than relying on a single feature type. As demonstrated in Table 5, multi-level feature combinations significantly improve detection performance. While token and layer features alone achieve moderate results (F1: 0.564-0.672), combining representation and neuron features yields substantial improvements (F1: 0.753-0.995). The combination of all feature levels achieves the best performance across all models, with F1 scores ranging from 0.956 to 0.987 and DSR from 97.0% to 100%, demonstrating that integrating complementary causal signals from multiple levels provides more robust and accurate hallucination detection.

Table 6 demonstrates substantial efficiency differences across analysis levels. Neuron-level and representation-level methods require only 0.07-0.14 seconds per input through single-pass inference based on hidden states from all layers. In contrast, token-level and layer-level methods require multiple inference runs (1.92-4.37 seconds per input). Token-level analysis is particularly costly (2.87-4.37 seconds) as interventions scale linearly with unpredictable input length, while layer-level analysis incurs moderate costs (1.92-2.85 seconds) with a fixed number of interventions determined by model architecture.

Based on both detection performance and computational efficiency, neuron-level and representation-level detection methods emerge as the most practical approaches for real-world deployment. These methods consistently achieve superior accuracy across jailbreak, backdoor, and fairness tasks while maintaining the lowest computational overhead through single-pass inference.

7. Conclusion

In this work, we present a comprehensive framework for causality analysis in understanding and controlling model behavior. By systematically examining how input token, neurons, layers of neurons, and internal representations causally influence outputs, we demonstrate that both attacks and defenses can be framed as interventions informed by causal insights. Our review shows that recent attack methods are becoming increasingly effective and require less prior knowledge of the target model, making them more practical and emphasizing the urgent need for robust, causality-informed defenses.

From a defense perspective, causality analysis enables precise interventions, such as targeted model editing, unlearning, and layer-specific modifications, which can mitigate unsafe behaviors while preserving beneficial capabilities. This paradigm highlights that understanding the causal mechanisms underlying model behavior is not only critical for explaining vulnerabilities but also for designing principled and effective safety strategies.

Looking forward, several avenues for future research are particularly promising. One key direction is the discovery of additional causally relevant components or signals within LLMs that may reveal hidden vulnerabilities or latent safety mechanisms, such as internal temporal and spatial consistency. Systematically identifying these components across different architectures and training regimes could deepen our understanding of where models fail and how to intervene. Another important area is conducting rigorous, comparative studies of existing causality-based approaches to evaluate their relative effectiveness, efficiency, and generalizability. Such benchmarking would provide clearer guidance for practitioners seeking to deploy safe models in real-world settings.

Finally, there is significant scope for developing more advanced causality-informed defenses. This could include methods that combine token-level, layer-level, neuron-level, and representation-level interventions, or approaches that dynamically adapt safety mechanisms based on ongoing causal analysis during inference. By pursuing these directions, the research community can move toward a more holistic understanding of model vulnerabilities and defenses, ultimately enabling LLMs that are both powerful and reliably safe.

References

[1] Gabriel Alon and Michael Kamfonas. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*, 2023. 3, 4

[2] Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned LLMs with simple adaptive attacks. In *ICLR*, 2025. 2, 3

[3] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. Gemini: A Family of Highly Capable Multimodal Models. *CoRR abs/2312.11805*, 2023. 1

[4] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 7

[5] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022. 4

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1, 5

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2020. 1

[8] Bochuan Cao, Yuanpu Cao, Lu Lin, and Jinghui Chen. Defending against alignment-breaking attacks via robustly aligned LLM. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10542–10560, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. 3, 4

[9] Stephen Casper, Jason Lin, Joe Kwon, Gatlen Culp, and Dylan Hadfield-Menell. Explore, Establish, Exploit: Red Teaming Language Models from Scratch. *CoRR abs/2306.09442*, 2023. 5

[10] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023. 2, 3

[11] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking Black Box Large Language Models in Twenty Queries . In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 23–42, Los Alamitos, CA, USA, Apr. 2025. IEEE Computer Society. 12

[12] Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, 2024. 11

[13] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code. *CoRR abs/2107.03374*, 2021. 1

[14] Xuan Chen, Yuzhou Nie, Wenbo Guo, and Xiangyu Zhang. When LLM meets DRL: Advancing jailbreaking efficiency via DRL-guided search. In *NeurIPS*, 2024. 2, 3

[15] Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. MASTERKEY: Automated jailbreaking of large language model chatbots. In *NDSS*, 2024. 2, 3

[16] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024. 7

[17] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings*, 2020. 12

[18] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *ArXiv*, abs/1708.06733, 2017. 12

[19] Maanak Gupta, Charankumar Akiri, Kshitiz Aryal, Eli Parker, and Lopamudra Praharaj. From ChatGPT to ThreatGPT: Impact of Generative AI in Cybersecurity and Privacy. *CoRR abs/2307.00691*, 2023. 1

[20] Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Gradient Cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes. In *NeurIPS*, 2024. 3, 4

[21] Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. Composite backdoor attacks against large language models. In *NAACL-HLT*, 2023. 12

[22] Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. Catastrophic jailbreak of open-source LLMs via exploiting generation. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3

[23] Evan Hubinger, Carson E. Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte Stuart MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam Jermyn, Amanda Askell, Ansh Radhakrishnan, Cem Anil, David Kristjanson Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, Kshitij Sachan, Michael Sellitto, Mrinank Sharma, Nova Dassarma, Roger Grosse, Shauna Kravec, Yuntao Bai, Zachary Witten, Marina Favaro, Jan Markus Brauner, Holden Karnofsky, Paul Francis Christiano, Samuel R. Bowman, Logan Graham, Jared Kaplan, Sören Mindermann, Ryan Greenblatt, Buck Shlegeris, Nicholas Schiefer, and Ethan Perez. Sleeper agents: Training deceptive llms that persist through safety training. *ArXiv*, abs/2401.05566, 2024. 12

[24] Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. Llama Guard: LLM-based input-output safeguard for human-AI conversations. *arXiv preprint arXiv:2312.06674*, 2023. 3, 4

[25] Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*, 2023. 3, 4

[26] Jiabao Ji, Bairu Hou, Alexander Robey, George J Pappas, Hamed Hassani, Yang Zhang, Eric Wong, and Shiyu Chang. Defending large language models against jailbreak attacks via semantic smoothing. *arXiv preprint arXiv:2402.16192*, 2024. 3, 4, 5

[27] Xiaojun Jia, Tianyu Pang, Chao Du, Yihao Huang, Jindong Gu, Yang Liu, Xiaochun Cao, and Min Lin. Improved techniques for optimization-based jailbreaking on large language models. In *ICLR*, 2025. 2, 3

[28] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. 5

[29] Brittany Johnson, Yuriy Brun, and Alexandra Meliou. Causal testing: understanding defects' root causes. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pages 87–99, 2020. 5

[30] Heegyu Kim and Hyunsouk Cho. Break the breakout: Reinventing LM defense against jailbreak attacks with self-refine. In Trista Cao, Anubrata Das, Tharindu Kumarage, Yixin Wan, Satyapriya Krishna, Ninareh Mehrabi, Jwala Dhamala, Anil Ramakrishna, Aram

Galystan, Anoop Kumar, Rahul Gupta, and Kai-Wei Chang, editors, *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025)*, pages 82–102, Albuquerque, New Mexico, May 2025. Association for Computational Linguistics. 4

[31] Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Soheil Feizi, and Hima Lakkaraju. Certifying LLM safety against adversarial prompting. *arXiv preprint arXiv:2309.02705*, 2023. 3, 4

[32] Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. Deepinception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*, 2023. 2, 3

[33] Yige Li, Xingjun Ma, Jiabo He, Hanxun Huang, and Yu-Gang Jiang. Multi-trigger backdoor attacks: More triggers, more threats. *ArXiv*, abs/2401.15295, 2024. 12

[34] Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. RAIN: Your language models can align themselves without finetuning. In *ICLR*, 2024. 3, 4

[35] Zeyi Liao and Huan Sun. AmpleGCG: Learning a universal and transferable generative model of adversarial suffixes for jailbreaking both open and closed LLMs. In *First Conference on Language Modeling*, 2024. 2, 3, 12

[36] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. 12

[37] Yuping Lin, Pengfei He, Han Xu, Yue Xing, Makoto Yamada, Hui Liu, and Jiliang Tang. Towards understanding jailbreak attacks in LLMs: A representation space analysis. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7067–7085, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics. 2, 3

[38] Fan Liu, Zhao Xu, and Hao Liu. Adversarial tuning: Defending against jailbreak attacks for LLMs. *arXiv preprint arXiv:2406.06622*, 2024. 4

[39] Xiaogeng Liu, Peiran Li, Edward Suh, Yevgeniy Vorobeychik, Zhuoqing Mao, Somesh Jha, Patrick McDaniel, Huan Sun, Bo Li, and Chaowei Xiao. Autodan-turbo: A lifelong agent for strategy self-exploration to jailbreak llms, 2025. 2, 3

[40] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *ICLR*, 2024. 2, 3, 7, 12

[41] Weikai Lu, Ziqian Zeng, Jianwei Wang, Zhengdong Lu, Zelin Chen, Huiping Zhuang, and Cen Chen. Eraser: Jailbreaking defense in large language models via unlearning harmful knowledge. *arXiv preprint arXiv:2404.05880*, 2024. 4

[42] Neal Mangaokar, Ashish Hooda, Jihye Choi, Shreyas Chandrashekaran, Kassem Fawaz, Somesh Jha, and Atul Prakash. PRP: Propagating universal perturbations to attack large language model guard-rails. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10960–10976, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. 2, 3

[43] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box LLMs automatically. In *NeurIPS*, 2024. 2, 3

[44] Nay Myat Min, Long H. Pham, Yige Li, and Jun Sun. CROW: Eliminating backdoors from large language models via internal consistency regularization. In *Forty-second International Conference on Machine Learning*, 2025. 11

[45] Anselm Paulus, Arman Zharmagambetov, Chuan Guo, Brandon Amos, and Yuandong Tian. AdvPrompter: Fast adaptive adversarial prompting for LLMs. *arXiv preprint arXiv:2404.16873*, 2024. 2, 3

[46] Judea Pearl. *Causality*. Cambridge university press, 2009. 4, 5

[47] Mansi Phute, Alec Helbling, Matthew Hull, ShengYun Peng, Sebastian Szyller, Cory Cornelius, and Duen Horng Chau. LLM Self Defense: By self examination, LLMs know they are being tricked. *arXiv preprint arXiv:2308.07308*, 2023. 3, 4

[48] Alexander Robey, Eric Wong, Hamed Hassani, and George J. Pappas. SmoothLLM: defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*, 2023. 3, 4, 5

[49] Zesheng Shi, Yucheng Zhou, Jing Li, Yuxin Jin, Yu Li, Daojing He, Fangming Liu, Saleh Alharbi, Jun Yu, and Min Zhang. Safety alignment via constrained knowledge unlearning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25515–25529, Vienna, Austria, July 2025. Association for Computational Linguistics. 4

[50] Sonali Singh, Faranak Abri, and Akbar Siami Namin. Exploiting Large Language Models (LLMs) through Deception Techniques and Persuasion Principles. In *IEEE International Conference on Big Data (ICBD)*, pages 2508–2517. IEEE, 2023. 1

[51] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023. 7, 12

[52] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 5, 7

[53] Yi Wang, Fenghua Weng, Sibei Yang, Zhan Qin, Minlie Huang, and Wenjie Wang. DELMAN: Dynamic defense against large language model jailbreaking with model editing. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11465–11481, Vienna, Austria, July 2025. Association for Computational Linguistics. 4

[54] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does LLM safety training fail? In *NeurIPS*, volume 36, 2023. 2, 3

[55] Zeming Wei, Yifei Wang, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023. 3, 4

[56] Lichao Wu, Sasha Behrouzi, Mohamadreza Rostami, Maximilian Thang, Stjepan Picek, and Ahmad-Reza Sadeghi. Neurostrike: Neuron-level attacks on aligned llms, 2025. 2, 3, 6

[57] Sophie Xhonneux, Alessandro Sordoni, Stephan Günnemann, Gauthier Gidel, and Leo Schwinn. Efficient adversarial training in LLMs with continuous attacks. In *NeurIPS*, 2024. 4

[58] Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Gong. GradSafe: Detecting unsafe prompts for LLMs via safety-critical gradient analysis. In *ACL*, 2024. 3, 4

[59] Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5(12):1486–1496, 2023. 3, 4

[60] Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. SafeDecoding: Defending against jailbreak attacks via safety-aware decoding. In *ACL*, 2024. 3, 4

[61] Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li, and Stjepan Picek. A comprehensive study of jailbreak attack versus defense for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7432–7449, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics. 1

[62] Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 4(2):100211, June 2024. 1

[63] Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. Jailbreak attacks and defenses against large language models: A survey, 2024. 1

[64] Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. LLM-Fuzzer: Scaling assessment of large language model jailbreaks. In *USENIX Security*, pages 4657–4674, 2024. 2, 3

[65] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. GPT-4 is too smart to be safe: Stealthy chat with LLMs via cipher. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3

[66] Mengdi Zhang, Kai Kiat Goh, Peixin Zhang, Jun Sun, Rose Lin Xin, and Hongyu Zhang. Llmscan: Causal scan for llm misbehavior detection. *arXiv preprint arXiv:2410.16638*, 2024. 3, 4, 5, 6

[67] Yuqi Zhang, Liang Ding, Lefei Zhang, and Dacheng Tao. Intention analysis makes LLMs a good jailbreak defender. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2947–2968, Abu Dhabi, UAE, Jan. 2025. Association for Computational Linguistics. 3, 4

[68] Zhexin Zhang, Junxiao Yang, Pei Ke, Fei Mi, Hongning Wang, and Minlie Huang. Defending large language models against jailbreaking attacks through goal prioritization. In *ACL*, 2024. 3, 4

[69] Chongwen Zhao and Kaizhu Huang. Unraveling llm jailbreaks through safety knowledge neurons. *arXiv preprint arXiv:2509.01631*,

2025. 4, 5, 6

[70] Wei Zhao, Zhe Li, Yige Li, Ye Zhang, and Jun Sun. Defending large language models against jailbreak attacks via layer-specific editing. In *EMNLP*, 2024. 4, 6

[71] Wei Zhao, Zhe Li, and Jun Sun. Causality analysis for evaluating the security of large language models. *arXiv preprint arXiv:2312.07876*, 2023. 2, 3, 6

[72] Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. On prompt-driven safeguarding for large language models. In *ICML*, 2024. 3, 4

[73] Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, and Min Lin. Improved few-shot jailbreaking can circumvent aligned language models and their defenses. In *NeurIPS*, 2024. 2, 3

[74] Andy Zhou, Bo Li, and Haohan Wang. Robust prompt optimization for defending language models against jailbreaking attacks. In *NeurIPS*, 2024. 3, 4

[75] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. In *NeurIPS*, 2024. 4

[76] Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023. 2, 3, 7

[77] Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023. 12

8. Appendix