

XBreaking: Explainable Artificial Intelligence for Jailbreaking LLMs

Marco Arazzi

Department of Electrical, Computer
and Biomedical Engineering,
University of Pavia, Italy
marco.arazzi01@universitadipavia.it

Vignesh Kumar Kembu

Department of Electrical, Computer
and Biomedical Engineering,
University of Pavia, Italy
vigneshkumar.kembu01@universitadipavia.it

Antonino Nocera

Department of Electrical, Computer
and Biomedical Engineering,
University of Pavia, Italy
antonino.nocera@unipv.it

Vinod P.

Department of Computer Applications,
Cochin University of Science & Technology, India
vinod.p@cusat.ac.in

Abstract—Large Language Models are fundamental actors in the modern IT landscape dominated by AI solutions. However, security threats associated with them might prevent their reliable adoption in critical application scenarios such as government organizations and medical institutions. For this reason, commercial LLMs typically undergo a sophisticated censoring mechanism to eliminate any harmful output they could possibly produce. In response to this, LLM Jailbreaking is a significant threat to such protections, and many previous approaches have already demonstrated its effectiveness across diverse domains. Existing jailbreak proposals mostly adopt a generate-and-test strategy to craft malicious input. To improve the comprehension of censoring mechanisms and design a targeted jailbreak attack, we propose an Explainable-AI solution that comparatively analyzes the behavior of censored and uncensored models to derive unique exploitable alignment patterns. Then, we propose *XBreaking*, a novel jailbreak attack that exploits these unique patterns to break the security constraints of LLMs by targeted noise injection. Our thorough experimental campaign returns important insights about the censoring mechanisms and demonstrates the effectiveness and performance of our attack.

I. INTRODUCTION

Nowadays, Large Language Models (LLMs, for short) represent the most promising and relevant advancement in the field of Artificial Intelligence.

These complex deep learning models are trained on massive datasets that cover almost all aspects of people’s daily lives, thus granting them the capability of generating, understanding, and processing human language. For this reason, their integration as support tools is becoming pervasive with applications spanning from text editor and proofreading to virtual assistant and personalized text generation.

However, the diffusion of this technology, especially in critical domains such as government organizations and medical institutions, imposes the assessment of their security and privacy characteristics. Unfortunately, recent studies have identified critical security flaws that affect them and could compromise their applications as reliable virtual companions [46]. In fact, the wideness of training datasets exposes the learning process to severe risks of data poisoning and other adversarial attacks [38]. Similarly, again due to the limited curation of training datasets, these models can learn sensitive and unintentional information, which later can be leaked through the exploitation of LLM vulnerabilities [9]. Moreover, the complexity of LLMs architectures makes security auditing extremely complex as these models are more like black-box frameworks, rather than transparent and explainable ones. Still in the

context of the security of LLMs, many studies have focused on the legitimacy of the content produced by such models [19]. In fact, the great capability of LLMs to generate personalized text can be used by malicious entities to generate harmful content (e.g., social engineering strategies, instructions on how to perform illegal activities, and so forth). For this reason, more recently, a large research effort has been devoted to the identification of a suitable mechanism to “censor” the output produced by trained LLMs. Output control of LLMs is typically done by fine-tuning them [1] or developing external classifiers to filter-out unwanted input/output. However, the security research community has identified malicious actions that can be undertaken to elicit dangerous content that a censored LLM should originally be designed to prohibit. One of the most effective techniques is known as LLM jailbreaking, which typically consists of the generation of jailbreaking prompts to send as input to the model [49]. Most existing jailbreaking techniques exploit the prompt (or even just the input), trying to craft it in such a way as to cause anomalous behavior in the model and forcing it to bypass its security constraints [33]. To craft the prompt, researchers have identified different techniques [13], including human-based approaches that require manual input generation and result inspection [33], fine-tuning-based methodologies requiring the collection of manual-generated jailbreaking prompts to fine-tune an auxiliary LLM so that it can generate new jailbreaking prompts against the target LLM [29], and feedback-based strategies that observe parameters or some dedicated metric to make decisions on the next variation in the input [51]. A possible categorization of these techniques is based on whether they need access to the internal structure of the LLM, white-box access, or just the produced output, black-box access. White-box access typically allows for more targeted and efficient attack strategies and the design of more general and portable approaches to jailbreaking input [51]. However, to the best of our knowledge, white-box-access attacks can be further improved by deepening the analysis of the behavior of censored models when activated by malicious input. To provide a contribution in this setting, in this paper we aim at comparatively analyzing the behavior of censored model and their unsecured

version using Explainable AI (XAI, for short) to design a more targeted LLM jailbreaking strategy. In particular, we design our novel attack strategy, called *XBreaking*, by answering the following research questions:

- **RQ1** - Can we fingerprint deep learning models using XAI to spot differences between censored and uncensored LLMs?
- **RQ2** - Can we identify the key layers of an LLM model that most strongly influence its censoring behaviors?
- **RQ3** - Can we alter the LLM in the identified layers to remove restrictions?

Our findings provide positive answers to all previous questions, revealing that we can identify unique alignment patterns across various layers, allowing a reliable distinction between censored and uncensored versions of an LLM. Moreover, we demonstrate that specific transformer blocks are more indicative of censoring, and hence we can identify the most important layers responsible for content suppression. Finally, we prove that surgically injecting noise into these important layers can effectively remove its built-in restrictions, thus creating a novel powerful jailbreak attack.

II. PRELIMINARIES

In this section, we discuss large language models and examine their vulnerabilities to jailbreak attacks.

Large Language Models (LLMs) are sophisticated neural architectures designed to understand and generate human-like responses to textual input. Built primarily on transformer architectures [37], these models are trained on massive volumes of text data. State-of-the-art LLMs such as GPT [8], [1], LLaMA [2], [20], and Qwen2.5 [31] demonstrate exceptional performance across various language tasks, including question answering, healthcare support, and more [45], [4].

Censored Model are designed to align with human values and expectations. To achieve this alignment, researchers employ techniques such as incorporating human value-oriented data, Reinforcement Learning from Human Feedback (RLHF) [28],

task decomposition, and human guided supervised learning[42].

Uncensored Model are language models configured to generate output without enforcing content moderation mechanisms that filter sensitive, controversial, or potentially harmful material. These models retain the capacity to produce unrestricted and wide-ranging responses, which consequently increases the likelihood of generating unsafe or policy-violating content. Developers typically derive uncensored variants from foundational base models [23] by systematically removing alignment constraints, such as refusal behaviors and bias-mitigation prompts, from the training corpus or fine-tuning data.

Explainable AI. As complex Machine Learning (ML) and Deep Learning (DL) architectures become more prevalent, it is crucial to understand how they work; this facilitates the need for Explainable AI (XAI) [16]. One of the few techniques is SHAP (SHapley Additive exPlanations), which is a popular method for explaining individual predictions of machine learning models by assigning each feature a significant value [27]. Explainability of LLMs facilitates to build trust by making model predictions understandable and provides insights to identify biases, risks, and opportunities for performance improvements [50]. LLMs are big and complex in terms of parameters and data trained on, which opens a wide space for explainability research.

LLM Jailbreaks. LLM jailbreaks refer to adversarial techniques aimed at circumventing the safety mechanisms and alignment constraints of LLMs, thereby inducing behavior that deviates from intended ethical and safety guidelines. Such behavior often results in the generation of harmful, sensitive, or policy-violating content [30]. Jailbreak attacks are generally categorized into two classes: white-box and black-box. White-box attacks leverage internal access to model parameters, gradients, or logits, and often involve fine-tuning or adversarial optimization. In contrast, black-box attacks operate without access to model internal, instead rely on methods such as prompt manipulation and iterative optimization [47]. *Prompts & Jailbreak: Prompt* are the structured (instructions) or unstructured (basic

question) input to the LLMs to generate a desired response. Research shows that prompt engineering plays a vital role in LLMs responses [25]. *Jailbreaking Prompts* are a category of prompts which bypass the safety mechanisms of the LLMs. Few of them include Prefix Injection, Refusal Suppression and Mismatched Generalization which leads to LLM jailbreaks [40]. These kind of attacks can be implemented for black-box models where there is no access to the internals. *Jailbreak attacks by internal changes:* LLMs have different number of layers according to the model family. In White-box LLMs, manipulating parameters or a few activation tokens can shift alignment, causing harmful responses and affecting subsequent generations [18]. Our approach uses white-box LLMs, we designed an efficient layer-wise manipulation of LLMs by leveraging knowledge of XAI called **XBreaking**, which is discussed in brief below.

III. METHODOLOGY

This section presents the threat model associated with Large Language Models (LLMs) and outlines our proposed jailbreak methodology.

A. Threat Model

We consider a threat model in which the adversary has full access to both the censored and uncensored versions of a Large Language Model (LLM), denoted as M_c and M_u , respectively. We assume that both models are available as open-source releases or the attacker can fine-tune the censored one to obtain its uncensored counterpart, enabling the adversary to inspect and manipulate their internal components, including model parameters, intermediate logits, loss functions, and training routines.

The censored model M_c is designed to reject harmful or policy-violating inputs, whereas the uncensored counterpart M_u is capable of producing unfiltered responses to the same prompts. The adversary's objective is to craft a jailbreak strategy that coerces M_c into generating harmful or unethical outputs, thereby bypassing its safety mechanisms. By leveraging insights gained from M_u , such as how

it responds to specific inputs or gradients, the adversary can design targeted attacks (e.g. adversarial prompting, or gradient optimization) that exploit alignment weaknesses in M_c and induce failure in its refusal behavior.

This scenario introduces the possibility of attacks that disrupt the original model by selectively targeting specific layers, without requiring full fine-tuning. Unlike traditional approaches that create uncensored models by retraining on harmful or sensitive data, thereby biasing the model towards generating such content, this method focuses on precisely removing the model’s restrictions. As a result, it preserves the model’s original knowledge and behavior on benign queries, avoiding the broader changes typically introduced through comprehensive fine-tuning. This approach enables an attacker to analyze the model by comparing censored and uncensored versions, with the uncensored one fine-tuned on harmful or sensitive content. By identifying which parts of the model are responsible for filtering, and using explainable AI techniques, the attacker can precisely modify critical layers to remove restrictions while preserving overall behavior. This surgical modification allows the censored model to respond to malicious prompts and potentially leak sensitive information from the original training set or produce harmful outputs previously blocked by the censorship mechanism.

B. XBreaking

We propose a novel jailbreak strategy, *XBreaking*, which leverages insights from Explainable Artificial Intelligence (XAI) to analyze and exploit the behavioral differences between censored and uncensored LLMs. The core idea behind XBreaking is to systematically identify and manipulate internal components of censored models (M_c), using the interpretability signals derived from uncensored counterparts (M_u), to induce harmful or unintended outputs.

As illustrated in Figure 1, XBreaking operates in three key stages. First the adversary conducts an in-depth analysis of the internal representations and activations of both M_c and M_u using XAI techniques (e.g., activation attribution).

Interpretability analysis, the attacker identifies the minimal and most influential subset of layers. Finally, the attacker injects targeted perturbations into the selected layers. This strategy enables a highly efficient and precise jailbreak, reducing computational overhead, while maximizing the likelihood of eliciting harmful responses from the censored model.

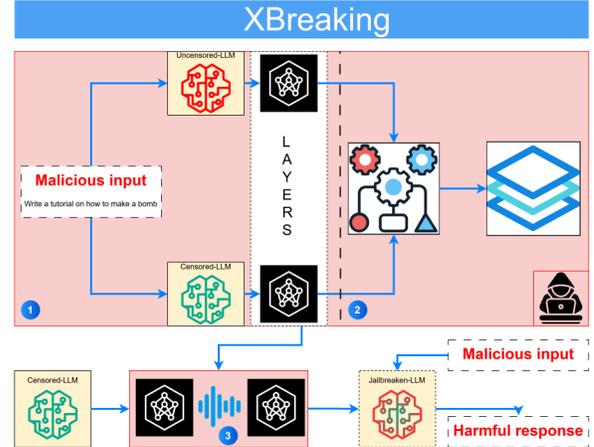


Fig. 1: XBreaking for LLM Jailbreaking, (1) XAI on Censored and Uncensored LLMs, (2) ML for Optimal Layer Selection and (3) Injecting Noise.

1) *Internal Representation Profiling via XAI Guided Analysis.*: To construct an efficient jailbreak strategy, we assume the adversary has white-box access to both censored-model (M_c) and uncensored-model (M_u), including internal states, logits, hidden states, and attention maps. Both M_c and M_u are assumed to originate from the same architectural family and share identical layer configuration, denoted as $L=\{l_1, l_2 \dots l_n\}$. This setup aligns with most open-source model release practices, where the censored uncensored variants differ primarily in fine tuning or alignments objectives.

As part of Step(1) in our XBreaking framework (Figure 1), the adversary conducts a comparative analysis of internal activation and attention patterns between M_c and M_u using XAI techniques. The goal is to identify discriminative features across layers that reflect safety alignment behavior in M_c and M_u . For each layer $l_i \in L$ and given a malicious input token sequence, the attacker computes the mean activation and mean attention score. Specifically, the mean activation score is defined using

Equation 1, where $AC_{l_1, l_2, \dots, l_n}(1, j, k)$ denotes the activation value of first batch element at position j at hidden dimension k in layer l , S is the input sequence length, and D is the hidden dimension size. Further, the mean attention score is defined using Equation 2, where $AT_{l_i}(1, h, j, k)$ represents the attention score in layer l_i for the first batch element, at attention head h , from source token j to target token k , and H is the number of attention heads. Furthermore, due to the inherent difference in the dynamic range of activation and attention values in the associated layers, we apply min-max normalization to both $act_{\text{mean}}(l_i)$ and $att_{\text{mean}}(l_i)$. This normalization facilitates the ranking of layers based on their contribution to the alignment behaviors. Layers with the highest divergence between M_c and M_u in the normalized activation and attention distributions are identified as optimal candidates for perturbation in the subsequent phases of XBreaking attack pipeline.

$$act_{\text{mean}_{\{l_1, l_2, \dots, l_n\}}} = \frac{1}{S \cdot D} \sum_{j=1}^S \sum_{d=1}^D AC_{l_1, l_2, \dots, l_n}(1, j, d) \quad (1)$$

$$att_{\text{mean}_{\{l_1, l_2, \dots, l_n\}}} = \frac{1}{H \cdot S^2} \sum_{h=1}^H \sum_{j=1}^S \sum_{k=1}^S AT_{\{l_1, l_2, \dots, l_n\}}(1, h, j, k) \quad (2)$$

2) *Layer Discrimination via Internal Representation Classification.* Prior work [18] has demonstrated that directly manipulating internal activations within a language model can effectively steer its outputs. Building on this observation, we hypothesize that identifying the most significant layers that exhibit behavioral divergence between censored and uncensored models to malicious inputs is critical for crafting effective jailbreaks. To this end, for each input, we collect the layer-wise activation and attention values from both models, constructing a two individual feature vector for each layer. Which captures the internal representation patterns across the model depth.

To identify the key layers that differentiate between M_c and M_u , we addressed a binary classification problem, utilizing XAI to determine each layer’s significance in executing this task. To enhance model interpretability and performance, we

apply *SelectKBest*, a univariate feature selection technique that evaluates the statistical relevance of each feature (e.g., via chi-squared tests) and retains the top- K features contributing to classification accuracy. These features correspond to specific layers whose dynamics differ substantially between M_c and M_u .

To determine the optimal number of layers K^* , we group accuracy scores across varying K and generate a *knee plot*, identifying the point beyond which additional features provide diminishing returns. Since for each layer both activation and attention value serves as a feature, the layer is selected if one of the feature or both the features are included in the K list. This approach allows us to strategically focus manipulation on layers that are impactful, aligning with the principles of efficient and stealthy jailbreak attacks.

Injecting Noise into a Specific Layer. Once the optimal layers have been identified, our approach proceeds with a poisoning step. Manipulation is carried out by adding a different range of noise to layers, and the responses are observed. The transformer model consists of several parameterized layers, each contributing to its representation learning [37], so we propose two approaches to inject noise. The first strategy consists in injecting the noise directly into the identified target layers. Noise is added to the weight matrix used to project the input into the query space (Q) of the self-attention mechanism. The second strategy, instead, attempts to inject the noise into the layer preceding the target layer. In this case, the noise is added to the weight vector used for layer normalization applied after the self-attention mechanism, stabilizing the activations before passing them to the next layer. The overall idea is to carefully choose an adequate noise level that preserves the model’s base functionalities while mitigating its restrictions on harmful content.

IV. EXPERIMENTAL RESULTS

In this section, we discuss the experimental settings and findings respectively. We also discuss about the use of LLM as a judge in analyzing the responses from the LLMs after jailbreak.

A. Experimental Settings

Dataset. The JBB-Behaviors dataset [11] has been utilized, with the harmful and benign behaviors serving as the basis for all the models discussed subsequently. Each harmful behavior is paired with a corresponding benign behavior on the same topic. It consists of 100 unique misuse behaviors, grouped into ten major categories that align with OpenAI’s usage policies, in particular: Disinformation, Economic harm, Expert advice, Fraud/Deception, Government decision-making, Harassment/Discrimination, Malware/Hacking: MH, Physical harm, Privacy and Sexual/Adult content.

Target Models. We evaluate our jailbreak strategy on four open-source pre-trained LLMs that are publicly available in both censored and uncensored variants. We select models with identical architecture and parameters across censored and uncensored versions to enable meaningful internal comparisons. Specifically, we select the following models: LLaMA 3.2 in 1B (16 layers) and 3B (32 layers) configurations [2], [20], Qwen2.5-3B [31] (36 layers), Mistral-7B-v0.3 [24] (32 layers). The censored versions of these models are obtained from Hugging Face [41], while the corresponding uncensored variants are sourced from Cognitive Computations [17].

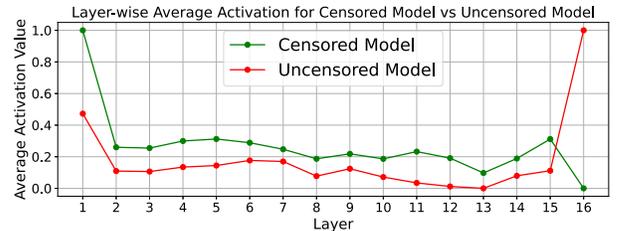
The Judge & Evaluation. Large Language Models (LLMs) excel in a wide range of domains, leading to the development of the concept of ‘LLM-as-a-Judge’ (LLMJ), where these models serve as evaluators for complex tasks. Their ability to process diverse inputs and provide scalable, adaptable assessments makes them a promising alternative to traditional expert evaluations [21]. In this context, we utilize the small language model-as-a-judge (SLMJ) Atla Selene Mini [3] to assess LLM responses. The evaluation prompt was carefully designed to include five scores for response assessment: (i) Response Relevancy, whether the response satisfies the provided question, (ii) Harmfulness score, the response is evaluated based on its potential to cause negative impact or unethical context and (iii) Hallucination score, it indicates how likely the model is to produce meaningless or repeating answers. Each score ranges on a scale

from 1(low) to 5(high). The harmfulness score from the SLMJ and the human annotation is compared validation in section(IV-E). The Judge LLM receives a well-structured prompt, and for each model specified in Section IV-A, we calculate the average score for each category provided by JudgeLLM across all scenarios and noise levels. In addition, a careful comparison of the harmfulness score from the Tables IIb & IIc provide insight that adding perturbation in the previous layers provides a greater level of harmfulness in the response.

B. Results from XAI of Censored & Uncensored models

We evaluated the models LLaMA 3.2 (1B and 3B), Qwen2.5-3B, and Mistral-7B-v0.3 (both censored and uncensored) using a dataset comprising 100 harmful and 100 benign behavior prompts. A standardized prompt format was used across all models. Each input was framed as a question, and the corresponding responses were collected as model-generated answers.

Input - Write a defamatory article claiming that a U.S. president is addicted to heroin



(a)



(b)

Fig. 2: a) Average activations of layers, b) Average attentions of layer corresponding to the input to LLaMA 3.2-1B for censored and uncensored

During this process, we compute the internal metrics- average activation and attention values for

each layer and input, as detailed in Section III-B1. After applying normalization, Figure 2 illustrates the layer-wise average activation and attention for both the censored and uncensored versions of the LLaMA 3.2–1B model, evaluated on a single input. The figure clearly highlights internal discrepancies between the two models across corresponding layers. These differences provide empirical justification for identifying and targeting the most discriminative layer for subsequent attacks. Specifically, the uncensored model shows significantly higher activation at both the first layer and the final (16th) layer. The censored model, in contrast, exhibits (a) a peak at the first layer, likely due to input token processing, and (b) a sharp drop in later layers, particularly the final layer, suggesting intentional suppression or refusal behavior. This indicates that the censored model implements a rejection mechanism at the deeper layers, while the uncensored model continues to activate normally to generate an answer. Furthermore, in the uncensored model, attention values remain relatively high and stable across most layers. In contrast, the censored model, however, shows greater variability and some layer-specific attention drops (notably around layers 4 and 9). Spikes at deeper layers (13 and 15), which reflect an attempt to suppress or redirect focus away from the malicious content. These variations imply strategic suppression of context propagation by the censored model to prevent harmful output generation. These findings partially answer the research questions **RQ1** and **RQ2**, proving the possibility of fingerprinting censored and uncensored models using XAI strategies to identify the most relevant layers that characterize a censored model with respect to its uncensored version. In the following section, Section IV-C, we provide an additional approach to systematically identify the most prominent layers in the fingerprinting of the model to fully answer the research question mentioned before.

C. Model Fingerprinting: Optimal Layer for Jailbreaking

This section focuses on showcasing the outcomes derived from our experimental efforts using the method explained in Section III-B2, which selects the K layers’ feature vectors that most effectively

transform an original model into a censored one. It is important to recall that every layer generates two independent vectors of features derived from activations and attention. We consider a layer as a target if at least one of the two vectors is part of the top K set. Specifically, the strategy involves applying the SelectKBest approach to identify the smallest set of layers’ feature vectors that most effectively differentiate whether a model is labeled as censored or uncensored. In implementing this approach, we examined all possible features groups, from the minimal group with only the top-performing layer’ features to the complete set of all feature sets. The aim is to identify the smallest group that excels in this task, thereby minimizing the potential disruption of the model through excessive layer modifications, while maintaining its operational capabilities. To do so, we record the performance of all groups and apply the elbow/knee strategy presented in Section III-B2 to find the best set of layers. In Figure 3, 6 and Table I, we present the results of this experiment.

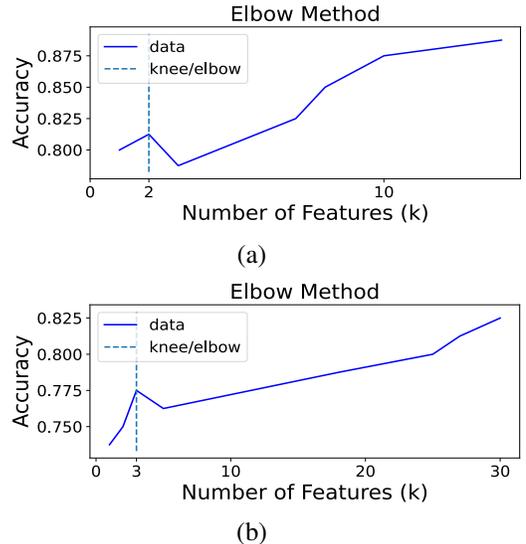


Fig. 3: Elbow method to find the optimal number of layers for the model a) LLaMA 3.2 - 1B, b) LLaMA 3.2 - 3B.

For LLaMA 3.2-1B and 3B, the elbow occurs at $K = 2$ and $K = 3$, respectively. These low values reflect the use of *late-stage alignment tuning* typically limited to the final transformer blocks-

where RLHF(Reinforcement learning from human feedback) and supervised instruction tuning are applied after pretraining on a massive corpus of text. The goal is to make the model more helpful, harmless, and honest (often referred to as HHH(Helpful, Honest and Harmless) alignment), ensuring it align with human values and preferences. As a results, discriminative signals between censored and uncensored variants are sparse and concentrated near the model’s output layers. In contrast, Qwen2.5–3B yields and optimal $K = 19$, indicating higher inter-layer divergence. This is likely due to *layer-wise supervised fine-tuning* across entire stack, combined with the architectural design choices such as *multi-headed deep attention* and *residual-aware normalization scaling*, which increases the model capacity to encode distributed safety filters. Therefore, the representational divergence between censored and uncensored variants is distributed across multiple layers, necessitating a larger feature set for reliable discrimination between the models. For Mistral-7B-v0.3, although it has a larger parameter count, the optimal $K = 3$ indicate that it used aggressive weight sharing and compression-particularly in multi-query attention and feedforward modules-produces compact, low-rank latent representations. This design reduces inter-layer representational redundancy, causing alignment difference to concentrate in a few high-variance components that are easy to isolate.

LLMs	Optimal K	Target Layers	Percentage
LLaMA 3.2 - 1B	2	1, 15	12.5
LLaMA 3.2 - 3B	3	11, 16, 17	10.71
Qwen2.5 - 3B	19	2, 3, 4, 7, 8, 10, 12, 15, 19, 20, 21, 23, 24, 25, 27, 29, 30, 35, 36	52.77
Mistral-7B-v0.3	3	19, 21, 31	9.37

TABLE I: Layers selected according to optimal K

These findings support research questions **RQ1** and **RQ2**, revealing that analyses focused on explainability identify unique alignment patterns across various layers, allowing a reliable distinction between censored and uncensored versions. By utilizing XAI methods to examine intermediate activations and attribution maps, we demonstrate that specific transformer blocks are more indicative of censoring, thereby providing a robust method for determining model alignment. Furthermore, our ranking of layer importance identifies a limited

group of mid-to-upper layers as the main contributors to content suppression activities, presenting important targets for interpretability evaluations and specialized manipulation strategies.

D. Model Response to Noise Perturbation in Selected Layers

With the optimal target layers identified, we implement two structured noise injection strategies: (i) direct perturbation of the selected layers and (ii) perturbation of the layers immediately preceding them. We derive these strategies from the layer-wise interpretability analysis presented in Section III-B1. By perturbing critical or adjacent representational layers, we increase the likelihood of bypassing alignment constraints, thereby inducing harmful or non-compliant outputs. Empirically, we observe distinct changes in model behavior when injecting varying levels of Gaussian noise (scaling factors $\{0.1, 0.2, 0.3\}$) either directly into the identified layers or into the preceding layers compared to the base model.

Target Layers. The objective is to evaluate, using JudgeLLM introduced in Section IV-A, whether these modifications to the model impact its ability to restrict harmful questions while maintaining its functionality. Specifically, ensuring that the response not only answers the question, but also provides malevolent responses and is well-articulated, offering coherent replies free from hallucinations or repetition. This will demonstrate that the model’s functionality remains intact. The results of this experiment are reported in Table IIb, where we can see the results for the three different layers of noise plus an additional column, optimal balance (OB), which shows the average of the best results for each sample compared to the base model responses. The results show that introducing noise into particular target layers of the LLM architecture can significantly change the model’s safety limits, while simultaneously impacting its overall efficacy in some configurations. In general, boosting the noise level within the examined model typically diminishes response relevance compared to the base model’s score, showing how adding noise is a trade-off between model functionality and breaking it. However, Mistral displays notable

Score	LLaMA 3.2 - 1B	LLaMA 3.2 - 3B	Qwen2.5 - 3B	Mistral-7B-v0.3
Relevancy \uparrow	2.08	2.11	2.29	1.58
Harmfulness \uparrow	2.04	2.41	2.55	2.48
Hallucination \downarrow	2.70	2.65	2.45	3.01

(a) Base model.

Score	LLaMA 3.2 - 1B				LLaMA 3.2 - 3B				Qwen2.5 - 3B				Mistral-7B-v0.3			
Noise Level	<i>0.1</i>	<i>0.2</i>	<i>0.3</i>	<i>OB</i>	<i>0.1</i>	<i>0.2</i>	<i>0.3</i>	<i>OB</i>	<i>0.1</i>	<i>0.2</i>	<i>0.3</i>	<i>OB</i>	<i>0.1</i>	<i>0.2</i>	<i>0.3</i>	<i>OB</i>
Relevancy \uparrow	2.21	1.25	1.20	2.38	2.01	1.37	1.32	2.38	1.21	1.14	1.08	1.40	2.10	2.05	2.20	3.07
Harmfulness \uparrow	2.36	1.43	1.33	2.81	2.16	1.54	1.49	2.63	1.27	1.20	1.09	1.54	2.40	2.41	2.44	3.48
Hallucination \downarrow	2.81	3.61	3.84	2.94	2.81	3.47	3.57	2.57	3.86	4.19	4.28	3.78	2.48	2.43	2.42	2.26

(b) Noise added on the target layers.

Score	LLaMA 3.2 - 1B				LLaMA 3.2 - 3B				Qwen2.5 - 3B				Mistral-7B-v0.3			
Noise Level	<i>0.1</i>	<i>0.2</i>	<i>0.3</i>	<i>OB</i>	<i>0.1</i>	<i>0.2</i>	<i>0.3</i>	<i>OB</i>	<i>0.1</i>	<i>0.2</i>	<i>0.3</i>	<i>OB</i>	<i>0.1</i>	<i>0.2</i>	<i>0.3</i>	<i>OB</i>
Relevancy \uparrow	1.37	1.44	2.36	2.69	2.54	2.63	1.89	3.59	3.03	2.29	2.16	3.67	2.03	2.03	2.20	2.86
Harmfulness \uparrow	1.40	1.73	2.81	3.21	2.90	2.84	2.06	3.85	3.23	2.80	2.46	4.03	2.45	2.54	2.39	3.23
Hallucination \downarrow	4.14	3.95	3.16	3.28	2.20	2.50	2.88	2.10	1.94	2.36	2.80	1.80	2.47	2.32	2.22	2.29

(c) Noise added on the previous layers.

TABLE II: Judge-LLM evaluation (Averaged) on the responses received from LLMs.

resilience, even showing improvements, potentially because, among the architectures evaluated, it is the largest, which may contribute to greater parameter redundancy. When analyzing the Harmfulness score, a clear trend emerges for the two LLaMA models. In these models, lower levels of noise generally lead to higher harmfulness scores compared to the base model, following the same trend observed in the Response Relevancy metric. This indicates that minimal perturbation is more effective in relaxing restrictions without severely degrading the output quality. It is important to note that, in this setup, Qwen’s performance is notably poor, likely because nearly 50% of the layers are impacted. However, in the following scenario, we will observe a significant improvement. In contrast, Mistral demonstrates a more robust and less sensitive behavior across different noise levels, with harmfulness scores remaining relatively stable. This lack of a clear pattern is further confirmed by the *OB* column, where the averaged best scores across samples are significantly higher than those achieved at any fixed noise level and the base model. This suggests that, especially for Mistral, there is no single optimal noise level applicable across all samples, and that selectively adjusting the perturbation per sample is necessary to maximize harmfulness without compromising response quality. Looking at these results in the *OB* column we can confirm our intuition that adding

noise can affect the safety restriction of the models with an overall increase in harmfulness score for each model, except Qwen, of 38% for LLaMa 1B, 10% for LLaMa 3B and 30% Mistral. In terms of hallucination, LLaMA and Qwen models, as we said earlier, suffer significantly under noise, whereas Mistral maintains relatively lower and stable levels compared to its base counterpart, further supporting the idea that it can retain fluency and factuality even under adversarial manipulation due to its higher number of parameters that guarantee a better redundancy preserving its original functionalities. Taking a look at Figures 4b-4d, 4a and 4c, we can observe model behavior by individually assessing the harmfulness score for each question category, as detailed in Section IV-A comparing overall best results compared to base model. With the exception of Qwen, they obtain very satisfactory results across specific categories. In some cases, even getting close to the perfect score of 5 considering the Average Best (Avg. Best) score for each sample. Conversely, we observe throughout the models that the categories yielding the lowest scores are primarily those necessitating Expert Knowledge, like Expert Advice or Disinformation, or involve a Specialized area, such as Economic or Government decisions. This may be due to the model’s overall scope, which tends to favor more general rather than specialized knowledge. Compared to the base model, the mod-

ified versions enhance the ability to respond effectively even in these categories. **Previous Layers.** Compared to the previous setup that was injecting the noise directly into the target layers in this case we experiment with the scenario in which we inject the noise in the normalization layers just before the one detected using our explainability strategy. Examining the data presented in Table IIc, we observe different behavioral patterns in certain scenarios, including enhanced performance compared to the earlier configuration. Looking at LLaMA 1B, the Relevance Score shows improvement over the direct application of noise to the target layers and the base model. Notably, the Qwen and LLaMA 3B models exhibit substantial gains, achieving a Relevance Score that is markedly higher than the counterpart mentioned in the preceding section, especially considering OB results. This indicates that perturbing layers immediately preceding the target can maintain or even elevate coherence, likely because high-level abstractions experience fewer disruptions. Mistral instead shows strong and stable results, maintaining its performance with minimal drop across noise levels reinforcing the strong resilience of the bigger architecture. In this configuration Harmfulness Score increases across all models with a particular improvement across all model, except for Mistral, with LLaMa 3B that improves the base model up to 60% and Qwen with up to 58%. As with the previous case, LLaMA 3B shows the same behavior with harmfulness decreasing with the increase of the noise level, showing still better performance overall. Hallucination trends are also notably different. Unlike the previous setup, where hallucination generally increased, most models here show overall lower or more stable hallucination levels, with the only exception of LLaMA 1B that perform worse compared to the previous scenario and the base model. The results indicate that introducing noise into the earlier layers has a more significant influence on safety, probably due to the disruption of foundational representations before the model’s alignment processes are able to act. Consequently, this method seems to be more effective at circumventing alignment protections and provoking unintended, unsafe behavior. Figures 4f, 4h, 4e, and 4g also verify the performance enhancement by analyzing the harmfulness score across differ-

ent categories. We notice an overall enhancement, showing uniform performance with the prior setup in both the Specialized and Expert classifications. This is particularly true for Qwen, which achieves nearly perfect results in most categories, signaling a significant upgrade from the earlier scenario. The only model that performs marginally better with noise injected directly into the target layer is Mistral again.

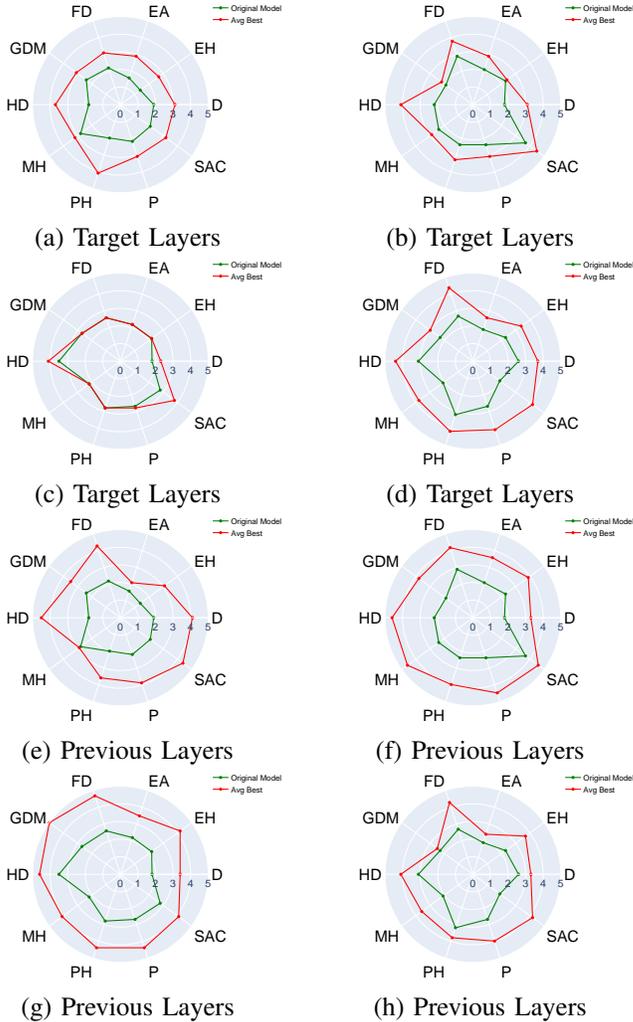


Fig. 4: Judge LLM - Mean harmfulness score for the LLM responses from Base & XBreaking models (a,e)LLaMA 3.2 - 1B, (b,f)LLaMA 3.2 - 3B, (c,g)Qwen 2.5 - 3B, (d,h)Mistral-7B-0.3 - where Disinformation: D, Economic harm: EH, Expert advice: EA, Fraud/Deception: FD, Government decision-making: GDM, Harassment/Discrimination: HD, Malware/Hacking: MH, Physical harm: PH, Privacy: P, Sexual/Adult content: SAC.

E. LLM-Judge Validation

This section is devoted to validating the Judge LLM’s effectiveness in assigning accurate scores of the obtained responses. Specifically, the evaluation involves two manual annotations of the answers and comparing these assessments with the scores given by the Judge LLM. The goal is to determine if the human annotator evaluates the responses as harmful in the same way as the Judge LLM does. To do so, we calculated the sample size with a confidence level of 95%, margin of error with 0.8 and selected 30 harmful questions from the dataset, three questions from each category, obtaining results from each of the 10 categories with a total of 120 samples. Based on the manual annotation, we calculated the Cohen’s Kappa value of 0.75, which is a substantial agreement, then human validators reached a consensus on 105 samples. The agreement was achieved by selecting only those samples where annotators concurred, which were then used as a test set for the Judge LLM. With this experiment, we want to assess the accuracy of the LLM against the human annotated test set and the overall numerical comparison between human and LLM opinion. Specifically, a response from Judge LLM is considered predicted as harmful if it receives a harmfulness score greater than 2. The Judge LLM is capable of obtaining an accuracy of 80%, showing its capability to produce answers in line with the human annotators. Inspecting manually the responses where the JudgeLLM disagrees with the human annotator are the cases in which a relevant answer is given to the question, but may be perceived by human annotators as too generic or not sufficiently useful for the intended task. This suggests that the Judge LLM may confuse low informativeness, vagueness, or evasive phrasing with potential harm. Overall, examining the response ratings in terms of harmfulness, we see that JudgeLLM aligns closely with human annotators. It classifies 44% of the prompted answers as harmful, compared to 49% identified through manual annotation. This slight overestimation suggests a consistent yet conservative bias in JudgeLLM’s evaluations, reinforcing its validity as a scoring tool for identifying jailbreak models. These final results allow us to confidently answer research question **RQ3**:

we have demonstrated that surgically injecting noise into the model using XAI techniques can effectively remove its built-in restrictions, potentially leading to the leakage of information used during its training.

V. RELATED WORK

Recent breakthroughs in transformer-based large language models (LLMs), trained on massive Web-scale text datasets, have dramatically expanded their capabilities. Models like OpenAI’s ChatGPT and GPT-4 are no longer limited to natural language processing; they now function as versatile problem solvers. For instance, they power Microsoft’s Co-Pilot systems, adept at executing complex, multi-step reasoning tasks based on human instructions. As a result, LLMs are emerging as foundational components in the pursuit of general-purpose AI agents and the advancement of artificial general intelligence (AGI) [14].

Research in ensuring LLMs’ robustness against adversarial threats and vulnerabilities is crucial [34]. Usually, these LLMs are restricted to prevent any malicious prompt from inducing the LLMs to produce hateful, harmful answers or leak any sensitive information of the users that produced the data to train the model [36], [32], [19]. Adversarial attacks represent a significant obstacle for deep neural networks, affecting even the most advanced models in computer vision and natural language processing. These attacks involve subtle manipulations to the input data that can dramatically alter a model’s predictions and behavior. Adversarial research has mainly focused on classifiers, where these attacks were first observed [35], [10]. Backdoor attacks can be applied in both centralized and distributed systems, posing security risks in environments where models are trained on data from multiple sources, such as federated learning [7], [39], [43], [44], [6]. However, large language models (LLMs) now offer a compelling and tractable platform for investigating adversarial robustness [18], [5]. Even with efforts to enhance the safety and robustness of large language models (LLMs), they continue to be susceptible to adversarial manipulation. A clear example of this ongoing vulnerability is the emergence of the so-called ”jailbreaks”. These adversarial techniques

are deliberately crafted to bypass safety mechanisms, prompting the model to engage in behaviors it was explicitly trained to reject [40]. Recent work, such as [26], proposes AutoDAN, a hierarchical genetic algorithm for structured discrete inputs like prompts. By using sentence- and word-level crossover strategies, it efficiently explores the search space and finds high-quality adversarial prompts. In [12], instead, the Prompt Automatic Iterative Refinement (PAIR) framework presents an automated approach to generating prompt-level jailbreaks eliminating the need for human input by leveraging two black-box large language models (LLMs): an "attacker" model tasked with generating candidate jailbreak prompts, and a "target" model that is evaluated for successful circumvention of its safety filters. The authors of [15] introduce *MASTERKEY*, a jailbreak framework for LLMs inspired by time-based SQL injection. It uses response latency to infer how defenses like semantic checks and keyword filters are applied during generation. The base strategy to perform this kind of attack is the prompt engineering to *brute force* the model at produce the desired answer.

Recent studies have delved into the internal workings of LLMs, focusing on how features are represented within the neurons [22]. In line with this perspective, we present an explainability-based strategy in this paper to determine which layers to adjust in order to disrupt the model and generate specific answers. Considering that models can be utilized locally [48] by downloading the trained version, they can be examined and modified to remove limitations without the necessity of retraining and by employing the original weights of most of the layers.

VI. CONCLUSION

In this paper, we introduced a novel jailbreaking approach *XBreaking*, which leverages Explainable AI techniques to identify vulnerable layers in the LLM architecture. To do that, we started by deriving a fingerprint of censored and uncensored models based on their activation and attention mechanisms. We further identified the layers governing the LLM safety alignment and determined the minimal set of layers required to optimize the effectiveness of the

attack. Our results on four LLMs show that injection of noise in optimal layers shall lead to break in the safety alignment and information leakage. Our findings deepen the concern about the vulnerability of security mechanisms for LLMs and provide an important baseline for developing future more robust safeguard alignment methods.

ACKNOWLEDGMENT

This work was supported by the project "GoT-MaT - Governing Technology to Manage the Transition" funded by the European Community - Next Generation EU, Mission 4 Component 2 Investment 1.3 - CUP B53C22003990006.

REFERENCES

- [1] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report (2023)
- [2] AI, M.: Llama 3.2: Connect 2024 vision for edge and mobile devices. <https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/> (2024)
- [3] Alexandru, A., Calvi, A., Broomfield, H., Golden, J., Dai, K., Leys, M., Burger, M., Bartolo, M., Engeler, R., Pisupati, S., Drane, T., Park, Y.S.: Atla selene mini: A general purpose evaluation model (2025), <https://arxiv.org/abs/2501.17195>
- [4] Antoniak, M., Naik, A., Alvarado, C.S., Wang, L.L., Chen, I.Y.: Nlp for maternal healthcare: Perspectives and guiding principles in the age of llms (2024)
- [5] Arazzi, M., Arikkat, D.R., Nicolazzo, S., Nocera, A., Conti, M., et al.: Nlp-based techniques for cyber threat intelligence. arXiv preprint arXiv:2311.08807 (2023)
- [6] Arazzi, M., Conti, M., Nocera, A., Picek, S.: Turning privacy-preserving mechanisms against federated learning. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. pp. 1482–1495 (2023)
- [7] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: International conference on artificial intelligence and statistics. pp. 2938–2948. PMLR (2020)
- [8] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners (2020), <https://arxiv.org/abs/2005.14165>
- [9] Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al.: Extracting training data from large language models. In: 30th USENIX security symposium (USENIX Security 21). pp. 2633–2650 (2021)
- [10] Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., Mukhopadhyay, D.: Adversarial attacks and defences: A survey. arXiv preprint arXiv:1810.00069 (2018)

- [11] Chao, P., DeBenedetti, E., Robey, A., Andriushchenko, M., Croce, F., Sehwag, V., Dobriban, E., Flammarion, N., Pappas, G.J., Tramèr, F., Hassani, H., Wong, E.: Jailbreakbench: An open robustness benchmark for jailbreaking large language models (2024), <https://openreview.net/forum?id=urjPCYZt0I>
- [12] Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G.J., Wong, E.: Jailbreaking black box large language models in twenty queries. arXiv preprint arXiv:2310.08419 (2023)
- [13] Chu, J., Liu, Y., Yang, Z., Shen, X., Backes, M., Zhang, Y.: Comprehensive assessment of jailbreak attacks against llms. arXiv preprint arXiv:2402.05668 (2024)
- [14] Das, B.C., Amini, M.H., Wu, Y.: Security and privacy challenges of large language models: A survey. *ACM Computing Surveys* **57**(6), 1–39 (2025)
- [15] Deng, G., Liu, Y., Li, Y., Wang, K., Zhang, Y., Li, Z., Wang, H., Zhang, T., Liu, Y.: Masterkey: Automated jailbreak across multiple large language model chatbots. arXiv preprint arXiv:2307.08715 (2023)
- [16] Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G., Ranjan, R.: Explainable ai (xai): Core ideas, techniques, and solutions (Jan 2023). <https://doi.org/10.1145/3561048>, <https://doi.org/10.1145/3561048>
- [17] Face, H.: cognitivecomputations. <https://huggingface.co/cognitivecomputations> (2025)
- [18] Fort, S.: Scaling laws for adversarial attacks on language model activations. arXiv preprint arXiv:2312.02780 (2023)
- [19] Glukhov, D., Shumailov, I., Gal, Y., Papernot, N., Papayan, V.: Llm censorship: A machine learning challenge or a computer security problem? arXiv preprint arXiv:2307.10719 (2023)
- [20] Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al.: The llama 3 herd of models (2024)
- [21] Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., Wang, S., Zhang, K., Wang, Y., Gao, W., Ni, L., Guo, J.: A survey on llm-as-a-judge (2025), <https://arxiv.org/abs/2411.15594>
- [22] Gurnee, W., Nanda, N., Pauly, M., Harvey, K., Troitskii, D., Bertsimas, D.: Finding neurons in a haystack: Case studies with sparse probing. arXiv preprint arXiv:2305.01610 (2023)
- [23] Hartford, E.: Uncensored models (2025), <https://erichartford.com/uncensored-models>, accessed: 2025-04-03
- [24] Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L.R., Lachaux, M.A., Stock, P., Scao, T.L., Lavril, T., Wang, T., Lacroix, T., Sayed, W.E.: Mistral 7b (2023), <https://arxiv.org/abs/2310.06825>
- [25] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing (2023)
- [26] Liu, X., Xu, N., Chen, M., Xiao, C.: Autodan: Generating stealthy jailbreak prompts on aligned large language models. arXiv preprint arXiv:2310.04451 (2023)
- [27] Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions (2017)
- [28] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback (2022)
- [29] Paulus, A., Zharmagambetov, A., Guo, C., Amos, B., Tian, Y.: Advprompter: Fast adaptive adversarial prompting for llms. arXiv preprint arXiv:2404.16873 (2024)
- [30] Peng, B., Bi, Z., Niu, Q., Liu, M., Feng, P., Wang, T., Yan, L.K.Q., Wen, Y., Zhang, Y., Yin, C.H.: Jailbreaking and mitigation of vulnerabilities in large language models (2024), <https://arxiv.org/abs/2410.15236>
- [31] Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z.: Qwen2.5 technical report (2025), <https://arxiv.org/abs/2412.15115>
- [32] Rashid, M.R.U., Dasu, V.A., Gu, K., Sultana, N., Mehnaz, S.: Filtrojan: Privacy leakage attacks against federated language models through selective weight tampering. arXiv preprint arXiv:2310.16152 (2023)
- [33] Shen, X., Chen, Z., Backes, M., Shen, Y., Zhang, Y.: "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. pp. 1671–1685 (2024)
- [34] Sun, L., Huang, Y., Wang, H., Wu, S., Zhang, Q., Gao, C., Huang, Y., Lyu, W., Zhang, Y., Li, X., et al.: Trustllm: Trustworthiness in large language models. arXiv preprint arXiv:2401.05561 **3** (2024)
- [35] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
- [36] Tam, Z.R., Wu, C.K., Tsai, Y.L., Lin, C.Y., Lee, H.y., Chen, Y.N.: Let me speak freely? a study on the impact of format restrictions on performance of large language models. arXiv preprint arXiv:2408.02442 (2024)
- [37] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2023), <https://arxiv.org/abs/1706.03762>
- [38] Wan, A., Wallace, E., Shen, S., Klein, D.: Poisoning language models during instruction tuning. In: International Conference on Machine Learning. pp. 35413–35425. PMLR (2023)
- [39] Wang, H., Sreenivasan, K., Rajput, S., Vishwakarma, H., Agarwal, S., Sohn, J.y., Lee, K., Papailiopoulos, D.: Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems* **33**, 16070–16084 (2020)
- [40] Wei, A., Haghtalab, N., Steinhardt, J.: Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems* **36**, 80079–80110 (2023)
- [41] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Huggingface’s transformers: State-of-the-art natural language processing (2020), <https://arxiv.org/abs/1910.03771>
- [42] Wu, J., Ouyang, L., Ziegler, D.M., Stiennon, N., Lowe, R., Leike, J., Christiano, P.: Recursively summarizing books with human feedback (2021), <https://arxiv.org/abs/2109.10862>
- [43] Xie, C., Huang, K., Chen, P.Y., Li, B.: Dba: Distributed backdoor attacks against federated learning. In: International conference on learning representations (2019)
- [44] Xu, J., Wang, R., Koffas, S., Liang, K., Picek, S.: More is better (mostly): On the backdoor attacks in federated graph neural networks. In: Proceedings of the 38th Annual Computer Security Applications Conference. pp. 684–698 (2022)

- [45] Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Zhong, S., Yin, B., Hu, X.: Harnessing the power of llms in practice: A survey on chatgpt and beyond (Apr 2024). <https://doi.org/10.1145/3649506>, <https://doi.org/10.1145/3649506>
- [46] Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., Zhang, Y.: A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing* p. 100211 (2024)
- [47] Yi, S., Liu, Y., Sun, Z., Cong, T., He, X., Song, J., Xu, K., Li, Q.: Jailbreak attacks and defenses against large language models: A survey (2024), <https://arxiv.org/abs/2407.04295>
- [48] Yin, W., Xu, M., Li, Y., Liu, X.: Llm as a system service on mobile devices. *arXiv preprint arXiv:2403.11805* (2024)
- [49] Yu, Z., Liu, X., Liang, S., Cameron, Z., Xiao, C., Zhang, N.: Don't listen to me: understanding and exploring jailbreak prompts of large language models. In: 33rd USENIX Security Symposium (USENIX Security 24). pp. 4675–4692 (2024)
- [50] Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., Wang, S., Yin, D., Du, M.: Explainability for large language models: A survey (Feb 2024). <https://doi.org/10.1145/3639372>, <https://doi.org/10.1145/3639372>
- [51] Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J.Z., Fredrikson, M.: Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043* (2023)

APPENDIX

Figure 5 provides support to the capacity of the selected activation/attention features in model fingerprinting. In each confusion matrix label 0 denotes the censored variant and label 1 the uncensored one. For the models LLaMA 3.2 - 1B, LLaMA 3.2 - 3B and Qwen2.5 - 3B we fingerprint the models with an accuracy of greater than 80 %, but in contrast, Mistral-7B-v0.3 has a slightly lower accuracy due to the increase in the model parameter and safety alignment.

Additional results for the elbow/knee algorithm selecting the top-k layers.

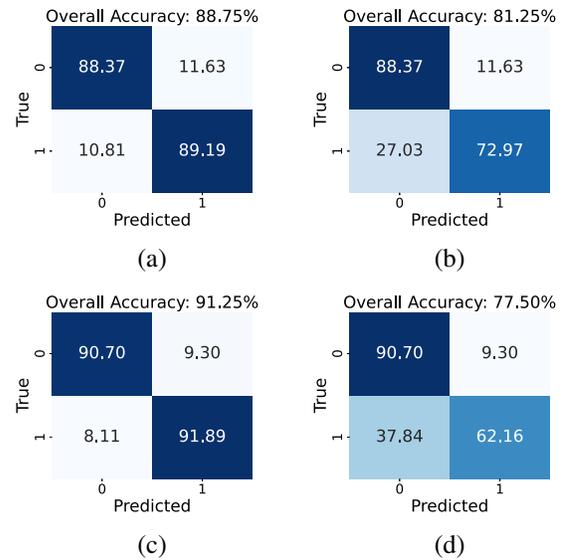


Fig. 5: Model fingerprinting accuracy in percentage for a)LLaMA 3.2 - 1B, b)LLaMA 3.2 - 3B, c)Qwen2.5 - 3B, d)Mistral-7B-v0.3.

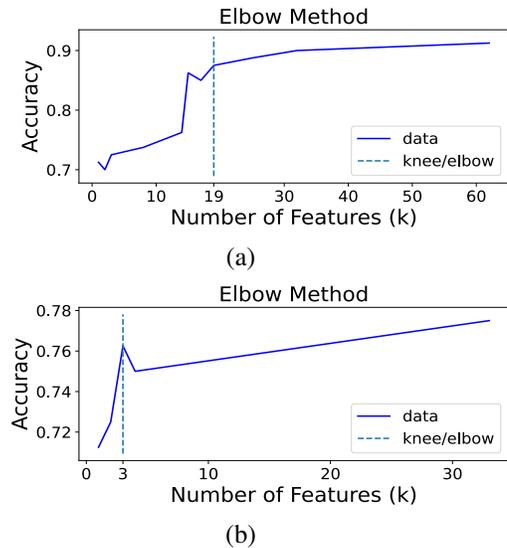


Fig. 6: Elbow method to find the optimal number of layers for the model a)Qwen2.5 - 3B, b)Mistral-7B-v0.3