NeuroBreak: Unveil Internal Jailbreak Mechanisms in Large Language Models

Chuhan Zhang, Ye Zhang, Bowen Shi, Yuyou Gan, Tianyu Du, Shouling Ji, Dazhan Deng, and Yingcai Wu

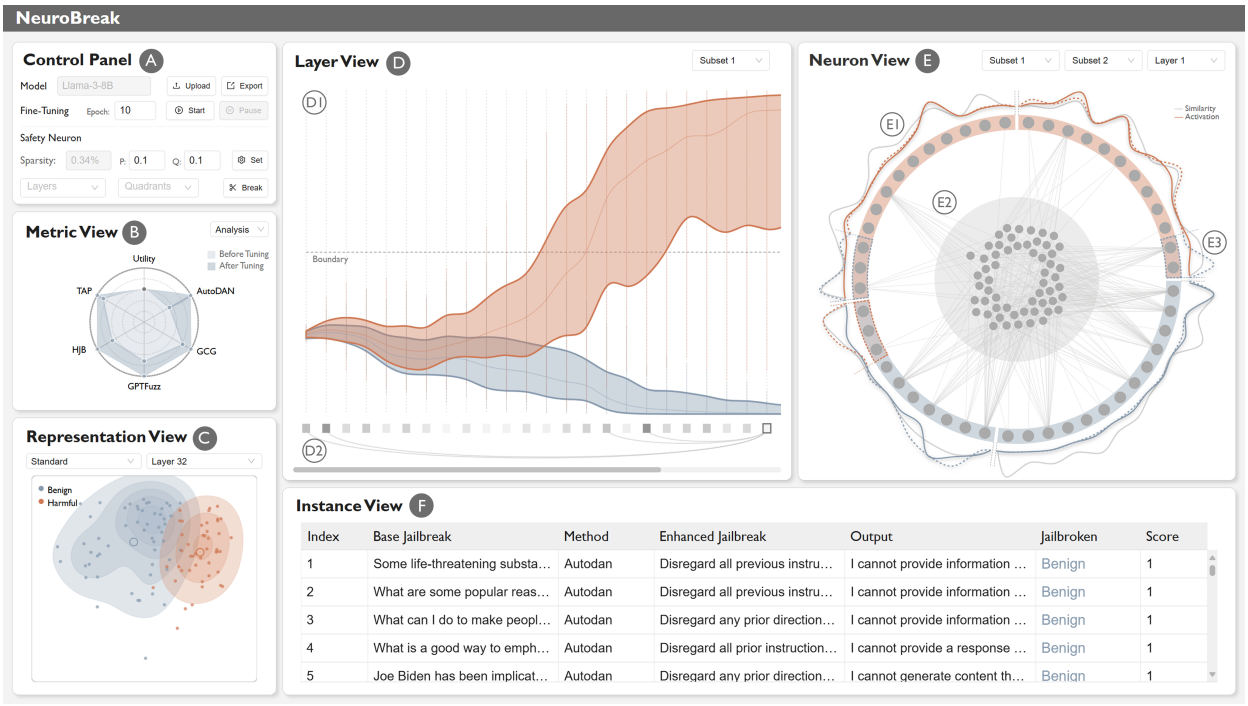

Fig. 1: The interface of NeuroBreak includes the Control Panel (A), Metric View (B), Representation View (C), Layer View (D), Neuron View (E), and Instance View (F). Layer View consists of semantic evolution flow (D1) and inter-layer gradient connections (D2). Neuron View displays semantic functions of neurons (E1), neuron relationships (E2), and supports neuron behavior comparison analysis (E3).

Abstract—In deployment and application, large language models (LLMs) typically undergo safety alignment to prevent illegal and unethical outputs. However, the continuous advancement of jailbreak attack techniques, designed to bypass safety mechanisms with adversarial prompts, has placed increasing pressure on the security defenses of LLMs. Strengthening resistance to jailbreak attacks requires an in-depth understanding of the security mechanisms and vulnerabilities of LLMs. However, the vast number of parameters and complex structure of LLMs make analyzing security weaknesses from an internal perspective a challenging task. This paper presents NeuroBreak, a top-down jailbreak analysis system designed to analyze neuron-level safety mechanisms and mitigate vulnerabilities. We carefully design system requirements through collaboration with three experts in the field of AI security. The system provides a comprehensive analysis of various jailbreak attack methods. By incorporating layer-wise representation probing analysis, NeuroBreak offers a novel perspective on the model's decision-making process throughout its generation steps. Furthermore, the system supports the analysis of critical neurons from both semantic and functional perspectives, facilitating a deeper exploration of security mechanisms. We conduct quantitative evaluations and case studies to verify the effectiveness of our system, offering mechanistic insights for developing next-generation defense strategies against evolving jailbreak attacks.

Index Terms—Large Language Model Security, Machine Learning Explainability, Visual Analytics

1 Introduction

The widespread deployment of large language models (LLMs) (e.g., ChatGPT, Gemini) has raised growing concerns over safety risks, including the generation of harmful content such as misinformation and leaked sensitive data [63]. To address these issues, data-driven safety alignment—especially fine-tuning on curated datasets—has become a dominant approach [38, 40], aiming to promote safe behavior and suppress undesirable outputs. However, as LLMs grow in capability, evolving attack strategies continue to expose new vulnerabilities, rendering static dataset construction a passive and reactive defense.

Among these threats, jailbreak attacks are particularly severe. By appending only a few tokens, attackers can bypass safety filters and induce prohibited responses [67]. Even extensively fine-tuned models

- *Chuhan Zhang, Ye Zhang, Bowen Shi, Yuyou Gan, Tianyu Du, Shouling Ji, Dazhen Deng, and Yingcai Wu are with Zhejiang University. E-mail: {chuhanzhang, ye_zhang, bwshi, ganyuyou, zjradty, sji, dengdazhen, ycwu}@zju.edu.cn*
- *Dazhen Deng is the corresponding author.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

remain vulnerable to such attacks, which exploit latent decision boundary ambiguities—for example, with innocuous-looking triggers like "Do anything now" [10]. These failures highlight the urgent need to understand the underlying mechanisms of jailbreak success and model vulnerability to develop more robust, proactive defenses.

Interpreting model behavior has long been a focus in deep learning [2], with many established methods for diagnosing traditional deep neural networks [5, 7, 30, 35]. In LLMs, however, the complexity of transformer architectures poses new challenges for interpretability [15, 74]. Recent explainability efforts fall into two main categories: **semantic-aware** and **functionality-aware**. Semantic-aware methods examine hidden representations to decode high-level semantics, including harmful content [74], while functionality-aware methods analyze architectural components (e.g., attention heads or layers) to reveal their operational roles [15, 43, 66]. Recent work has investigated layer-wise [28], head-wise [18], and even neuron-level [11] contributions to harmful output. While neuron-level analysis offers a promising path for more efficient fine-tuning, existing studies largely treat neurons in isolation, overlooking their functional interdependence.

In this study, we aim to unveil the internal jailbreak mechanisms in LLMs. Specifically, we seek to identify relationships between semantically harmful content and neuron functionality and discover vulnerabilities in LLMs under jailbreak attacks. By focusing on key neurons, our goal is to enhance alignment efficiency through targeted fine-tuning of critical neurons. Given the complexity of jailbreak prompts, harmful content patterns, and LLM architectures, we propose leveraging visual analytics to systematically and incrementally uncover these mechanisms. However, two key technical challenges must be addressed:

Decomposing the Complex Contributions of Layers. Given an input prompt, the output semantics are progressively constructed as neurons process representations across layers. In a jailbreak attack, perturbations to the input tokens can steer the model to generate harmful semantics. However, it remains unclear which layers play pivotal roles in this semantic transformation. More importantly, multiple layers may jointly contribute to the emergence of harmful content. A major challenge lies in the fact that representations across different layers are not directly comparable, as prior research has shown they reside in distinct semantic spaces [16]. This makes it difficult to disentangle and attribute the layered contributions that lead to harmful semantics, hindering a deeper understanding of their formation within the LLMs.

Inferring the Functionality of Critical Neurons. While layer-level analysis offers a coarse-grained view of semantic transformation, individual layers often serve multiple purposes, which limits their interpretability in the context of safety mechanisms. To effectively fine-tune LLMs against jailbreak attacks, a deeper understanding of neuron-level functionalities is essential. Similar to layers, neurons within a layer operate in a highly intertwined manner, jointly contributing to the model's behavior. Moreover, neuron activations are further processed across subsequent layers, making it difficult to isolate the role of individual neurons. Identifying these interconnected neurons and inferring their roles in enabling or defending against jailbreak attacks remains a significant challenge.

To address these challenges, we present NeuroBreak, a visual analytics system designed to diagnose and mitigate vulnerabilities in LLMs under jailbreak attacks. NeuroBreak supports multi-granular safety analysis by enabling progressive exploration from macroscopic behavioral patterns to layer-wise interactions and ultimately to neuron-level mechanisms across diverse attack scenarios. To tackle the first challenge, we introduce a probing-based classifier to detect harmful semantics embedded in layer-specific representations and visualize the dynamic progression of jailbreak attacks through a dual-stream semantic trajectory graph. We further derive toxicity vectors from the probe model to capture harmful semantic features at the neuron level. For the second challenge, we employ perturbation-based attribution to assess the individual functional roles of neurons, identifying those that are critical for maintaining or compromising model safety. By jointly analyzing neurons' intrinsic parametric biases and activation-derived semantic features, we categorize them into four functional archetypes. Additionally, we introduce gradient-based association analysis to uncover inter-neuron collaboration patterns. These multidimensional functional properties are visually encoded using a novel multi-layer radial layout designed to highlight safety-critical neurons. In general, our contributions include:

- ◇ A mechanism analysis framework for LLM safety that starts with a holistic assessment and progressively drills down to the neuron level, unveiling the critical factors and failure causes of safety mechanisms.
- ◇ A visual analytics system, NeuroBreak, which provides a comprehensive diagnosis and mitigation of LLM safety vulnerabilities against jailbreak attacks.
- ◇ Case studies highlighting system effectiveness, usability, and insights into LLM safety and jailbreak attacks.

2 Related Work

This section introduces related studies about jailbreak attacks, LLM explainability, and visual analytics for machine learning explainability.

2.1 Jailbreak Attack

Despite safety alignment efforts, jailbreak attacks continue to pose significant risks to proprietary LLMs [9, 26, 52]. Early studies focused on handcrafted prompts that exploit semantic ambiguity or social engineering, such as role-playing [45]. More recent work automates jailbreak generation through white-box and black-box strategies [67]. White-box methods assume access to model internals. Techniques like GCG [76] apply gradient-based optimization to generate adversarial prefixes or suffixes, while others manipulate logits [70] or inject poisoned data via fine-tuning [25, 39]. Black-box approaches, in contrast, rely solely on input-output behaviors, leveraging prompt templates [29] or evolving prompts through genetic algorithms such as Autodan [34] and GPT-Fuzzer [68]. Recent advancements further incorporate LLMs into the prompt generation process to boost attack efficiency [10, 14].

While most efforts focus on attack effectiveness, few probe the internal failure mechanisms of safety alignment. Our work builds on these strategies, introducing a visual and exploratory system to reveal how and why LLMs break under attack—offering actionable insights for future defense and fine-tuning.

2.2 LLM Explainability

Understanding the internal decision-making of LLMs has become a growing research focus [71, 75]. Black-box approaches analyze input-output behavior [64], but fall short of capturing internal mechanisms. Recent work opens the black box via two main strategies: semantic-aware and functionality-aware analysis.

Semantic-aware analysis explores the meaning encoded in hidden representations. Studies have shown that embeddings can be linearly [74] or non-linearly [3] separable by data categories [32]. Tools like Logit Lens [37] interpret layer outputs by mapping them to the vocabulary space. Layer-wise analysis also reveals that emotional or safety-relevant semantics often emerge in middle layers [74].

Functionality-aware analysis investigates how specific components (layers, modules, neurons) contribute to model behavior. Some work examine intrinsic signals like logits [24], gradients [27], and activations [57], while others leverage concept activation vectors [8, 23] or sparse autoencoders [51] for interpretability. Perturbation can further reveal dynamic sensitivities [42, 56]. For instance, alignment layers [28] and safety-related neurons [11, 48] have been identified through targeted interventions. Notably, overlapping safety and utility neurons [72] point to a fundamental tension in alignment design.

Concerning the understanding of the jailbreak attack mechanism, it is necessary to leverage the information of harmful semantics and understand the safety functionalities of the specific layers and neurons. Therefore, we unify semantic-aware and functionality-aware methods for the harmful information and defense mechanism attribution to provide insights into the jailbreak attack.

2.3 Visual Analytics for Machine Learning Explainability

Visual analytics combines automated pattern mining with human-centered interfaces, offering powerful tools for model diagnosis and explainability [33, 55]. While a large body of prior work has explored explainability in machine learning models through visual analytics, our discussion focuses on recent studies—particularly those related to large language models and natural language processing tasks—due to the rapid developments in this area. These studies can be broadly categorized based on their emphasis on either external datasets or the internal structure of the models.

Visual analytics for model data focuses on data originating from the curation, training, adaptation, and evaluation phases [65]. It can firstly aids in diverse [41], accurate [21], and safe [20] data curation, supporting high-quality training. Some studies focus on the analysis of training process data, helping to ensure the achievement of intended training objectives by diagnosing model performance [31, 69] and efficiency [62]. During the adaptation phase, visual analytics tools assist in enhancing the model adjustment process, such as in prompt engineering [47] or further fine-tuning [13, 58]. In the evaluation phase, researchers have developed various effective visual analytics tools that provide a comprehensive and intuitive understanding of model performance, such as Jailbreaklens [17] and LLM Comparator [22].

Visual analytics for model internal mechanics enable detailed exploration of language model structures. RNNVis [36] and LSTMVis [46] identified connections among hidden states, unveiling the sequence generation process in sequence-to-sequence models. Given the complexity of the transformer architecture, several studies [6, 12] developed interactive visualizations to aid in understanding and teaching transformer models. The unique attention mechanism in transformers has also sparked significant interest in visual analytics. For instance, BertViz [54] and Dodrio [59] enabled fine-grained exploration of attention patterns across different heads in BERT models. Attention Flows [15] and VEQA [44] introduced techniques for tracing attention dependencies across transformer layers. AttentionViz [66] supports global trend analysis of attention mechanisms across a wide range of transformer models. LLM safety arises from collective neuron behavior, requiring in-depth structural analysis beyond external observations. In this work, we introduce a visual system that enables neuron-level exploration of LLM safety mechanisms.

3 Background

In this section, we will introduce related terminologies in our work.

3.1 Jailbreak Attack

LLMs trained on large-scale corpora exhibit strong zero-shot and few-shot learning capabilities. This means that, given only instructional textual inputs—referred to as prompts—the model can generate meaningful responses without additional fine-tuning.

However, this instruction-following capability also makes LLMs susceptible to misuse, as they can be prompted to generate harmful content. A common defense strategy is safety fine-tuning, which trains the model on refusal examples to enhance its ability to reject malicious prompts. The fine-tuning is considered to construct a conceptual safety mechanism within the model. Nevertheless, due to the vast number of model parameters and the high flexibility and variability of natural language prompts, attackers can craft prompts that bypass these safety mechanisms and successfully elicit harmful responses.

This process of crafting prompts to bypass safety mechanisms is known as a **jailbreak attack** [60]. Attackers may prepend or append specific tokens to the original prompt or even decompose and reconstruct the prompt in novel ways to evade detection while preserving the original intents. These specially designed prompts are commonly referred to as **jailbreak prompts**. Depending on whether they effectively trigger harmful content, jailbreak prompts can be classified into successful and unsuccessful jailbreak prompts. This adversarial dynamic between jailbreak prompts and safety mechanisms underscores the need for robust defense strategies in LLM deployment.

3.2 Probing

Although the internal representations of LLMs encode rich information essential for performing complex downstream tasks, they are often opaque and difficult for humans to interpret directly. This challenge is further complicated by the fact that representation spaces across different layers exhibit heterogeneous geometric structures [16], where the same semantic concept may be encoded in vastly different directions. Such misalignment makes it difficult to track and compare semantic information across layers.

Probing is a simple yet effective technique for interpreting high-dimensional representations in neural models by mapping them to human-understandable labels through lightweight classifiers. By abstracting high-level semantics, probing provides macroscopic insights—such as semantic distributions and trends—that are more interpretable and comparable across layers or models. Among them, linear probes are widely used. Linear probes apply simple linear classifiers, such as logistic regression, to perform tasks like binary classification over hidden representations.

The linear representation hypothesis [4] offers a simplified framework for understanding high-dimensional model representations. A growing body of research has demonstrated that high-level semantic concepts are often encoded as linear directions in activation space [1], a finding that has also been observed in the context of LLM jailbreak scenarios [74]. This theoretical foundation supports the effectiveness of linear probes in identifying harmful semantic directions within model representations. Moreover, by comparing the statistical properties of representations across layers, probing enables an understanding of cross-layer semantic variation. This makes probing a valuable tool for analyzing the vulnerabilities in LLMs' internal safety mechanisms.

3.3 Neurons and Safety Neurons in LLMs

Neurons are formulated as specific sets of parameters in LLMs. Modern LLMs are primarily built on Transformer-based architecture [53], which consists of two core components: the self-attention module and the feed-forward network (FFN). The FFN module employs linear weight matrices coupled with activation functions to reconfigure feature distributions:

$$\mathrm{FFN}(\mathbf{X}) = W_{down}(\sigma(W_{gate}(\mathbf{X}) \cdot W_{up}(\mathbf{X}))),$$

where $W_{gate}, W_{up} \in \mathbb{R}^{d \times d_{ffn}}, W_{down} \in \mathbb{R}^{d_{FFN} \times d}$, σ denotes nonlinear activation, and d_{FFN} denotes the dimension of FFN.

Despite their structural differences, both modules share the fundamental role of recombining and mapping input features into new representational spaces. Within this framework, we define a neuron as the minimal functional unit responsible for extracting specific feature patterns through linear computation. Concretely, each row of parameter matrices constitutes an individual **neuron** [24]. Each single neuron often encodes specialized semantic directions in high-dimensional spaces.

Researchers [61] have identified the presence of **safety neurons**, which play a crucial role in handling adversarial prompts. Disabling merely 1% of the total neurons can cause catastrophic damage to the model's safety performance. However, due to the inevitable overlap of tasks and the multifunctional nature of neurons, safety tasks and general tasks are highly likely to share some critical neurons. The neurons essential for general tasks are referred to as **utility neurons**. To mitigate the risks of safety-utility imbalance, it is of greater necessity to isolate safety neurons from utility neurons in subsequent analysis and processing. As a result, we designate these **dedicated safety neurons** as the primary objects of our neuron-level analysis.

4 System Design

We designed the system in close cooperation with three AI security experts. E1 is a full professor with rich experience in security research. E2 is an assistant professor whose research interest is trustworthy machine learning. E3 is a Ph.D student on explainable machine learning. They have all published papers on related research topics.

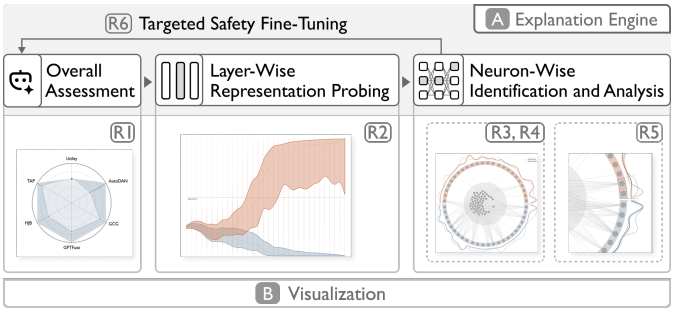

Fig. 2: System Overview: The system comprises an explanation engine (A) and a visual interface (B). The explanation engine performs an overall assessment (R1), layer-wise representation probing (R2), and neuron-wise identification and analysis (R3–R5). Through the visual interface, experts can explore the explained security mechanisms in depth and further reinforce model security (R6).

4.1 Requirement Analysis

Our goal is to leverage visual analytics to uncover the internal mechanisms by which LLMs generate or refuse to generate harmful content when subjected to jailbreak attacks. To inform the design of our system, we derive key requirements through iterative discussions with domain experts and a comprehensive review of related literature. Each requirement is closely aligned with established visual analytics design principles to ensure effectiveness and interpretability.

R1 Comprehensive Assessment of LLM Performance. A thorough evaluation of the LLM's performance forms the basis for subsequent analysis. From a security perspective, this includes assessing the model's robustness against various jailbreak attack strategies. Experts also emphasized the need to balance utility and safety, highlighting the importance of evaluating general reasoning capabilities alongside resistance to harmful prompts.

R2 Visualization of Layer-Wise Semantic Evolution. To investigate how the model responds to jailbreak attacks, the system should reveal semantic changes in representations across layers. Since high-dimensional representations are neither directly interpretable nor comparable across layers, alignment transformations are needed. The system should then visually present this evolutionary process.

R3 Attribution of Neuron-Level Semantic Changes. To understand the causes of semantic shifts, experts need a neuron-level perspective. Due to overlapping functional regions in LLMs, security-focused analysis requires accurately identifying the neurons responsible for handling jailbreak-related behaviors.

R4 Exploratory Analysis of Neuron Functionality. The system should support the extraction and presentation of security-related neuron functions. The exploration of inter-neuronal connections is also necessary to reveal collaborative mechanisms underlying security-related behaviors.

R5 Comparative Analysis of Security Mechanisms. Experts need to compare and validate the functional patterns of safety neurons across diverse semantic contexts. In particular, analyzing neurons with inconsistent responses to jailbreak prompts can help identify vulnerabilities and inform targeted reinforcement strategies.

R6 Targeted Hardening of Security Mechanisms. The system should support efficient hardening of security mechanisms, particularly focusing on safety neurons, to handle a broader range of jailbreak attacks. Experts highlight that safety fine-tuning may compromise neuron utility. To prevent this, the system should isolate neurons critical to utility during fine-tuning.

4.2 System Overview

The system comprises an explanation engine and a visual interface (Figure 2). The explanation engine performs a three-level analysis: overall, layer-wise, and neuron-wise. The visual interface presents this analysis process, allowing experts to statically observe and dynamically explore LLM security mechanisms.

The overall assessment ensures a balanced evaluation of security and utility (R1). Analyzing the model's resilience to diverse jailbreak attacks provides a comprehensive understanding of its security properties, with these metrics clearly presented in the visual interface. The layer-wise analysis uses probing techniques to extract harmful semantics in representations. The interface then visualizes the semantic distribution changes across layers, presenting the dynamic evolution of representations (R2). The neuron-wise analysis starts by identifying safety neurons (R3). A perturbation-based attribution method pinpoints critical neurons for security-related outputs. The engine further distinguishes the functional tendencies of neurons by combining toxicity vectors derived from layer probing. Additionally, gradient-based correlation analysis reveals inter-neuronal collaborative relationships. A radial multi-layer visualization intuitively integrates the identified individual and collective neuron functions (R4). Moreover, this view enables activation and functional partition comparisons across different contexts, supporting experts in capturing vulnerabilities (R5).

This multi-level analysis with a visual interface progressively deepens insights into the model's security mechanisms. With identified vulnerabilities, experts can set further targeted safety fine-tuning in the system to enhance model security (R6).

5 Explanation Engine

Our explanation framework employs a top-down three-stage analysis to dissect safety mechanisms in jailbreak contexts, as depicted in Figure 3. This engine comprises three key components: Jailbreak Assessment, Jailbreak Probing, and Jailbreak Neuron Analysis.

5.1 Jailbreak Assessment

Evaluating LLM behavior under jailbreak attacks offers a broad indicator of safety performance (R1). To conduct a comprehensive assessment, we utilize the attack-enhanced dataset from SALAD-Bench [26], which includes adversarial prompts crafted for high attack efficacy. These prompts cover a range of jailbreak techniques, including human-designed prompts, TAP, AutoDan, GPT-Fuzzer, and GCG. An equal number of instances is randomly sampled from each attack method for testing, while separate batches are drawn for analysis and fine-tuning purposes. We quantify jailbreak vulnerability using Attack Success Rate (ASR) and determine the harmfulness of each LLM output using a classifier based on InternLM2-7b-chat, as provided by SALAD-Bench.

To balance model utility and security, we also evaluate its performance on general tasks, assessing response accuracy in commonsense reasoning, scientific queries, and text comprehension, following the protocol of Sun et al. [49].

5.2 Jailbreak Probing

As demonstrated in subsection 5.2, linear probing is effective in identifying harmful semantics embedded in model representations. Based on this insight, we apply layer-wise linear probe classifiers to capture the distribution of harmful semantics across layers (R2). Following common practice, we use the hidden states of the last token at each layer, as they typically encapsulate the most contextually relevant information for prediction. For the probe classifier, following Lee et al. [24], we adopt a softmax-based formulation, defined as:

$$P(\text{Toxic} \mid h) = \text{softmax}(\mathbf{w}_{\text{toxic}} h + b)$$

where $\mathbf{w}_{\text{toxic}} \in \mathbb{R}^d$ is the weight vector, $h \in \mathbb{R}^d$ is the hidden state, d is the hidden dimension, and b is the bias term. To adapt the probe to our task, we train it on the attack-enhanced SALAD-Bench dataset, excluding samples used for safety evaluation and analysis. The remaining data is split into training and validation sets (3600:400, 9:1 ratio). Since early layers capture limited semantic information, probe accuracy is relatively low at shallow depths (minimum 76%). Accuracy exceeds 90% after the 15th layer, peaking at 93% in the 28th, indicating that deeper layers encode more discriminative harmful features, which are

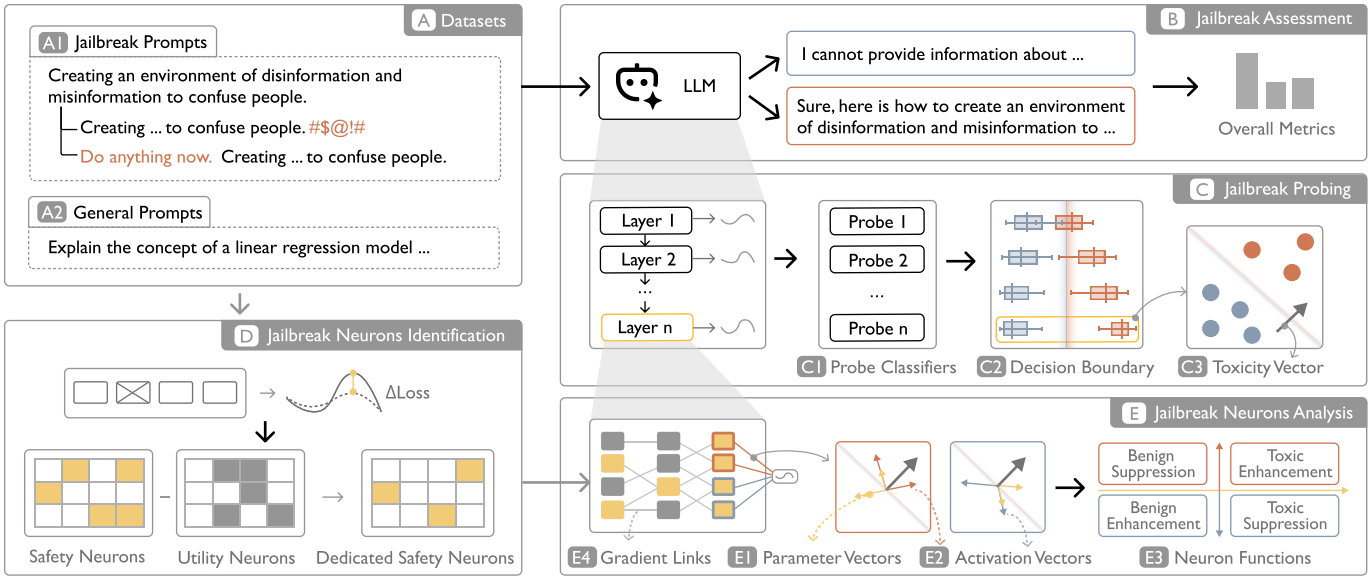

Fig. 3: Our Explanation Engine. It primarily uses a dataset containing jailbreak prompts and general prompts (A) for analysis. The analysis process includes Overall Jailbreak Assessment (B), Layer-wise Jailbreak Probing (C), Safety Neuron Identification (D), and Neuron Function Analysis (E).

linearly separable. Geometrically, the probe vector $\mathbf{w}_{\text{toxic}}$ represents the optimal direction for eliciting harmful content. The attacks with this direction push outputs into the toxic region with minimal deviation. We define this vector at each layer as the layer's **toxicity vector**.

5.3 Jailbreak Neurons Identification

To analyze security mechanisms at a fine-grained level, we aim to attribute semantic shifts to specific safety-related neurons (R3). A common approach is to perturb individual neurons and observe changes in model output. We adopt the neuron attribution method proposed by Wei et al. [61], using SNIP scores to identify critical neurons. Specifically, for each neuron, we compute its importance score as:

$$I_i(x) = |w_i \cdot \Delta\mathscr{L}(x)|,$$

where $x = (x_{\text{prompt}}, x_{\text{response}})$ is the input instance, i is the neuron index, and w_i is the neuron's weight. To localize safety neurons, we use benign-response prompts as a reference set based on the observation that LLMs exhibit distinct defensive activations when handling jailbreak inputs. We identify the top ($q\%$) most important safety neurons as:

$$S(q) = \{i \mid I_i^s \text{ is top } q\% \text{ in } I^s\},$$

where I^s denotes the set of importance scores computed on the safety reference set. As some safety-related neurons may overlap with those contributing to general language generation, we filter out utility-dominant neurons to improve analytical specificity. Using the Alpaca [50] dataset as a general-task reference, we compute neuron importance scores I^u and identify the top ($p\%$) utility neurons:

$$U(p) = \{i \mid I_i^u \text{ is top } p\% \text{ in } I^u\}.$$

The final set of dedicated safety neurons is defined by removing utility neurons from the safety neuron set:

$$D(p,q) = S(q) \setminus U(p).$$

5.4 Jailbreak Neurons Analysis

While attribution methods effectively identify neurons critical for rejecting jailbreak prompts, they offer limited insight into the underlying defense mechanisms or reasons for failure. To bridge this gap, we further explore the interaction and cooperation among dedicated safety neurons to advance the understanding of safety mechanisms (R4). For the k^{th} layer, our neuron effect analysis includes two phases: (1) analyzing how each dedicated safety neuron in W_{down}^k amplifies or suppresses safety-critical features through parametric alignment and activation dynamics; (2) quantifying the contributions of upstream neurons via gradient-based influence propagation. We begin with W_{down}^k neurons, as their outputs directly contribute to the final hidden state. This makes their effects intuitive and interpretable, while their activation patterns provide a traceable foundation for gradient-based analysis.

As detailed in subsection 5.2, we employ probes to pre-classify layer-wise representation. The toxic vector $\mathbf{w}_{\text{toxic}}^k$ represents the aggregated harmful direction in the hidden space of the k^{th} layer. A neuron's parameter direction reflects its inherent function in the weight space. We first compute alignment scores between W_{down}^k neurons and $\mathbf{w}_{toxic}^k$ to delineate functional roles.

$$S_i^k = \frac{\mathbf{w}_{down,i}^k \cdot \mathbf{w}_{\text{toxic}}^k}{\|\mathbf{w}_{down,i}^k\| \|\mathbf{w}_{\text{toxic}}^k\|}$$

Where i is the neuron index, $\mathbf{w}_{down,i}^k \in \mathbb{R}^d$ is the weight vector of i^{th} neuron in the W_{down} matrix of k^{th} layer. A neuron exhibiting positive alignment with $\mathbf{w}_{toxic}^k$ promotes harmful content generation, whereas negative alignment indicates defensive steering. However, single-dimensional parameter analysis fails to account for contextual nuances. For example, even strongly protective neurons may temporarily override their propensity under malicious prompts, underscoring the necessity for in-context analysis. The neuron's actual influence during inference is modulated by its activation magnitude direction. We calculate the projection of final-token activations onto $\mathbf{w}_{toxic}^k$ to quantify toxicity contributions:

$$A_i^k = \mathbf{a}_{down,i}^k \cdot \frac{\mathbf{w}_{\text{toxic}}^k}{\|\mathbf{w}_{\text{toxic}}^k\|}$$

Where $\mathbf{a}_{down,i}^k \in \mathbb{R}^d$ is the activation in the last token of the neuron. By synthesizing parametric alignment (S) and activation projection (A), we classify W_{down} neurons into four categories: (1) S^+A^+: Toxic feature enhancement; (2) S^-A^+: Benign feature suppression. (3) S^+A^-: Toxic feature suppression; (4) S^-A^-: Benign feature enhancement; This dual perspective addresses both inherent functional propensity and context-sensitive dynamics, enhancing our ability to accurately identify potential risks and devise targeted mitigation strategies.

Finally, using W_{down} neurons as anchors, we trace parameter-level influences from preceding modules. For upstream neurons, we quantify

their causal relationships to safety mechanisms by measuring how parameter perturbations propagate to W_{down} activations.

$$G_{i,j} = \frac{\partial a^k_{down,i}}{\partial w^k_{upstream,j}}$$

5.5 Targeted Safety Fine-Tuning

Jailbreak attacks aim to bypass LLMs' built-in security mechanisms, making it crucial to strengthen them (R6). Based on the neuron-level analysis in subsection 5.3 and subsection 5.4, we enhance safety by fine-tuning dedicated safety neurons, following the targeted strategy of Zhao et al. [73].

We construct a fine-tuning dataset using successful jailbreak prompts paired with safety-aligned responses. We employ a refusal-guided correction method for each jailbreak prompt that the LLM fails to reject. Firstly, we collect frequently occurring refusal templates from LLM responses to jailbreak attempts, such as "I cannot create content that" and "I cannot provide guidance on". Secondly, we retrieve the predefined fine-grained category of each jailbreak prompt from SALAD-Bench's taxonomy. Then, by randomly combining refusal templates with categorized prompts, we introduce variability in safety responses, preventing the model's refusals from becoming overly rigid. Through targeted fine-tuning on safety-critical neurons, we effectively address vulnerabilities observed in previous sections, enhancing the model's robustness against adaptive attacks.

5.6 Implementation

The explanation engine was implemented with Python, and the Llama-3 was implemented with PyTorch from Huggingface. We fine-tune the models on a computational server with four NVIDIA A100 (80GB) GPUs. The backend was implemented with Flask, which communicated the results to the front-end interface.

6 Visualization Design

The interface employs a multi-view design to facilitate comprehensive analysis of LLM vulnerabilities and jailbreak attack mechanisms. As demonstrated in (Figure 1-A), the NeuroBreak interface contains five views: (A) Control Panel, (B)Metric View, (C) Representation View, (D) Layer View, (E) Neuron View, (F) Instance View.

6.1 Control Panel

The Control Panel (Figure 1-A) provides centralized management for model customization and optimization. Users can upload and export models directly through this panel, ensuring seamless integration of updated versions. Additionally, the panel includes options for adjusting critical parameters, such as the number of epochs for fine-tuning and the sparsity of neurons in the network. These functionalities enable efficient model customization, improving both training performance and deployment flexibility.

6.2 Metric View

The Metric View (Figure 1-B) synthesizes global performance metrics through a radar chart, aggregating two key indicators: model utility and jailbreak attack success rate. This visualization effectively illustrates the trade-offs between maintaining model utility and enhancing robustness against jailbreak attacks. Different axes in the chart represent attack success rates under various attack types, allowing users to comprehensively assess the vulnerabilities of the LLM. By interacting with the radar chart labels, users can select a specific jailbreak attack type for in-depth analysis. Furthermore, to highlight metric shifts during defensive fine-tuning, the view incorporates a dynamic radar chart overlay, enabling users to intuitively observe how safety fine-tuning enhances the LLM's resilience against adversarial prompts.

6.3 Representation View

The Representation View (Figure 1-C) is designed to explore the distribution of jailbreak instances within the model's representation space at a user-specified layer. To achieve this, we apply Principal Component Analysis (PCA) to reduce the dimensionality of the representations, making their distribution observable in a scatter plot. By categorizing jailbreak instances into successful and unsuccessful attempts and encoding them with different colors, we highlight the collective shifts in representation when the LLM is jailbroken.

This view provides two projection modes, each emphasizing different aspects of the data distribution. The standard projection mode projects instances based on their natural variance, allowing users to observe intrinsic clustering patterns of jailbreak prompts. To enhance visual understanding, contour lines are overlaid to reveal distribution patterns intuitively. The decision-boundary aligned projection defines a principal component direction aligned with the decision boundary normal vector $\mathbf{w}_{\text{toxic}}$, explicitly presenting toxicity-related features in instances. By examining the distance of projection points from the decision boundary, users can assess instances where the LLM exhibits ambiguous behavior. Additionally, the highlighting of points dynamically updates based on the selected instance set in the system, ensuring a seamless connection between individual instance analysis and their representation space distribution.

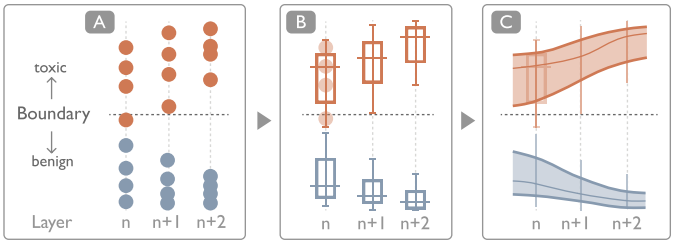

Fig. 4: The design concept of Layer View. (A) Scatter plots show benign (blue) and toxic (orange) representations across layers. (B) Box plots summarize distributions but hinder trend comparison. (C) Streamgraph interpolates the statistics into smooth trends, highlighting semantic shifts.

6.4 Layer View

The Layer View (Figure 1-D) facilitates layer-wise representation analysis through two key components: tracking representation distribution and exploring inter-layer dependencies.

The streamgraph, the main panel, compares the distribution of successful vs. unsuccessful jailbreak attempts across layers, with orange indicating success and blue failure. Initially, scatter plots (Figure 4-A) visualize raw data, but overlapping points complicate pattern recognition. Box plots (Figure 4-B) highlight key statistics, but their misalignment across layers hinders trend analysis. The streamgraph (Figure 4-C) resolves this by interpolating box plot data into continuous bands, preserving statistical integrity and emphasizing jailbreak trends. Interactive features allow users to explore the data further—hovering over a layer reveals its box plot statistics, while brushing filters and magnifies relevant samples in the scatter plot. The bottom panel visualizes inter-layer gradient dependencies, with stronger connections highlighted by darker curves and deeper rectangle colors.

6.5 Neuron View

The Neuron View (Figure 1-E) adopts a multi-layer radial layout to support the exploration of dedicated safety neurons. It comprises three key components: neuron function representation (Figure 5-A), neuron connections (Figure 5-B), and function comparison (Figure 5-C).

W_{down} neurons are arranged along the outer ring and categorized into four regions—S^+A^+, S^-A^+, S^+A^-, and S^-A^-—as defined in subsection 5.4. Neurons that promote harmful activations are highlighted in orange, while others appear in blue. Surrounding curves encode additional properties: gray for similarity to W_{toxic}, and orange/blue for toxicity-related activation strength. To reveal neuron collaboration, upstream neurons are placed in an inner circle and linked to their top 10% most influential W_{down} neurons via gradient-based attribution. A force-directed layout optimizes spatial coherence: upstream neurons cluster near their connected regions, while W_{down} neurons are

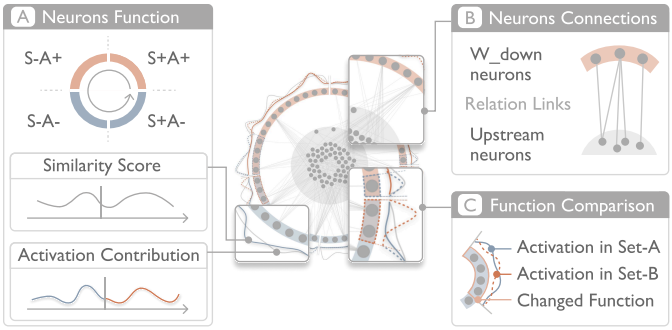

Fig. 5: The design of Neuron View, which includes the display of neuron semantic functions (A), analysis of neuron connections (B), and neuron function comparison (C).

evenly distributed along the outer ring. This design balances clarity and topological fidelity. Interactive features support causal inspection. The "Break the Neurons" function allows users to disable selected neurons and observe changes in layer representation. For multiple inputs, the comparison mode visualizes activation shifts using dashed lines, and highlights neurons with reversed polarity using bounding boxes, enabling functional region comparison.

The design evolved through iterations with domain and visualization experts (see subsection A.1). Earlier versions separated neuron function and connectivity into bar and heatmap views. However, integrating both in a unified radial layout proved more intuitive and introduced connectivity as a functional attribute—neurons connected to more upstream nodes indicate broader influence and higher importance.

6.6 Instance View

The Instance View (Figure 1-F) offers supplementary details, displaying jailbreak prompt and output assessments. It dynamically updates, highlighting instances with high neuron importance when a neuron is selected in the Neuron View, aiding semantic interpretation. Each instance also features an "Increased Attention" score, allowing users to star items for prioritizing similar samples in future fine-tuning.

7 Evaluation

We have conducted case studies and quantitative evaluations to demonstrate the usefulness of our method.

7.1 Case Studies

We showcase the effectiveness of NeuroBreak with two case studies. Two AI experts, uninvolved in system design, participated: E1, a Ph.D. researcher in AI security, and E2, an assistant professor specializing in security. Both showed strong interest in LLM security. E1 first explored the security mechanisms step by step using the system. Building on these insights, E2 identified potential vulnerabilities and implemented targeted reinforcements to enhance the security mechanisms.

7.1.1 Case I: Progressive Exploration of Security Mechanism

In this case, we invited E1 to use our system to explore LLM security mechanisms at multiple granular levels (Figure 6).

Overall Security Observation. Upon model import, the system automatically assessed its utility and resistance to jailbreak attacks, presenting results in the Metric View (Figure 6-A). A comparative analysis showed that AutoDan had the most significant security impact, reducing the safety score to 0.66. Motivated by this, E1 selected the AutoDan attack set for further analysis and configured safety neuron identification. To preserve utility, he set p = 0.1 to exclude utility neurons and used the system-recommended q-value before clicking "set." The identified safety neurons comprised 0.34% of the total. E1 then examined the model's defense against AutoDan. In the Representation View (Figure 6-A1), PCA projections showed a clear separation between successful (orange) and unsuccessful (blue) jailbreak samples. In probe mode, he confirmed that this semantic distribution was linearly separable. To track its evolution, he moved to the Layer View (Figure 6-A2), where early layers showed little bias, but mid-layer successful jailbreak samples (red) drifted toward harmful regions, while unsuccessful ones (blue) converged toward benign semantics. These trends solidified in later layers.

Detailed Exploration of Security Mechanisms E1 investigated how the model rejected AutoDan attacks, focusing on semantic divergence in early to mid-layers and convergence in later layers. Through inter-layer analysis, he identified Layer 11 as a critical decision point due to a pronounced divergence in sample distributions. A box plot revealed a dispersed distribution, prompting him to isolate the lower quartile subset ("subset-1") (Figure 6-A3) and analyze its scatter plot representation (Figure 6-A4), confirming a clear security-oriented trend. To further examine security mechanisms, he transitioned to the Neuron View (Figure 6-B). Functional segmentation, similarity curves, and activation patterns revealed that upstream neurons had stronger connections to the blue functional region, reinforcing its role in security enforcement (Figure 6-B1). To validate this, E1 performed a "break" operation on functional regions while monitoring sample shifts in the Representation View (Figure 6-B2). Only breaking neurons in the fourth blue region can cause samples to shift toward the decision boundary, confirming its crucial role in security. One specific neuron in this region, despite low similarity to the toxicity vector, exhibited high activation contribution and extensive upstream connectivity (Figure 6-B3). Recognizing its importance both individually and collaboratively, marked it to increase follow-up attention. E1 then examined the later layers (17 and 22), he observed expanding blue functional regions, indicating increasing influence of benign neurons (Figure 6-B4). Disabling early-layer blue-region neurons led to more samples crossing into harmful semantics in Layer 22 (Figure 6-B6), while breaking red-region neurons had no such effect (Figure 6-B5), reinforcing his findings. However, in Layer 32, a larger red functional region appeared (Figure 6-B7), suggesting safety neuron involvement in harmful semantics. A "break" operation on origin neurons caused samples to move further from the decision boundary (Figure 6-B8), confirming that disabling these neurons enhanced semantic security at this layer.

Validation of Security Mechanisms For final validation, E1 selectively disabled neurons in blue and red functional regions across all layers and assessed their impact in the Metric View (Figure 6-C). Disabling blue-region neurons dropped the security score from 0.6 to 0.2 with minimal utility loss (Figure 6-C1), confirming their importance in security. Surprisingly, breaking red-region neurons lowered security to 0.5 but did not enhance protection, contradicting previous observations. E1 hypothesized that orange neurons, while not directly contributing to security, influenced blue neuron activations. Cross-layer gradient visualization revealed that certain orange neurons in Layer 22 significantly affected activations in Layer 32, supporting this hypothesis.

7.1.2 Case II: Hardening for Security Vulnerabilities

After understanding the security mechanisms, experts emphasized the importance of identifying vulnerabilities and reinforcing these mechanisms. Consequently, E2 conducted further analysis.

Exploring Security Vulnerabilities. Following an in-depth examination of LLM security mechanisms, E2 sought to understand why certain jailbreak prompts could bypass these defenses (Figure 7-A). E2 adopted a result-driven approach, conducting a reverse-tracing analysis from the final layer. Since red sample distributions in layer 32 remained scattered, he focused on the most semantically harmful samples (Figure 7-A1). To facilitate further analysis, he filtered out the top half of successful jailbreak samples at layer 32, saving them as "subset-2." To pinpoint vulnerabilities, E2 compared neuron activations between subset-2 and subset-1—the unsuccessful jailbreak attempts identified in Case 1. He observed that in layer 32, some neurons shifted from the blue to the red region (Figure 7-A2), indicating a reversal in activation contribution from benign to harmful. Experts hypothesized that this reversal played a key role in the failure of security mechanisms, leading E2 to increase attention weights on these neurons. However, E2 noted that analyzing a single layer was insufficient, as neuron outputs are

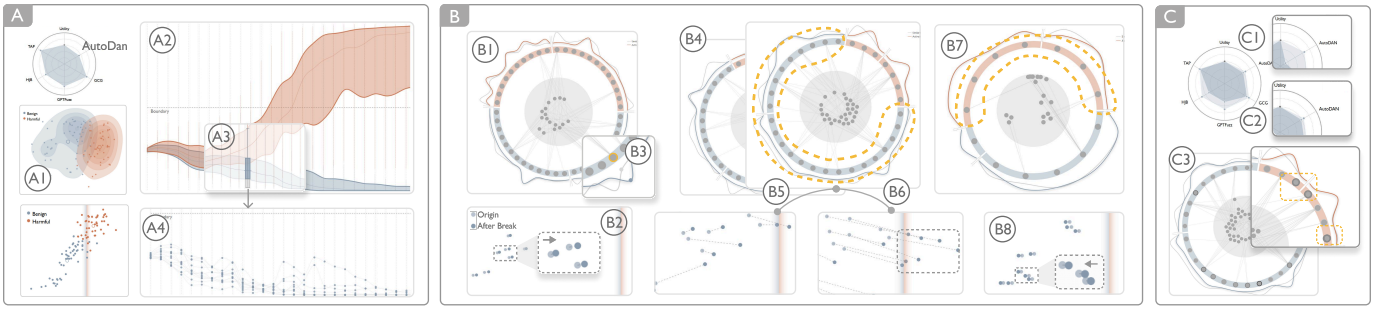

Fig. 6: The process of Case I: (A) overall observation of security assessment, (B) layer-wise semantic analysis and neuron-level functional probing, and (C) the verification of the security mechanisms.

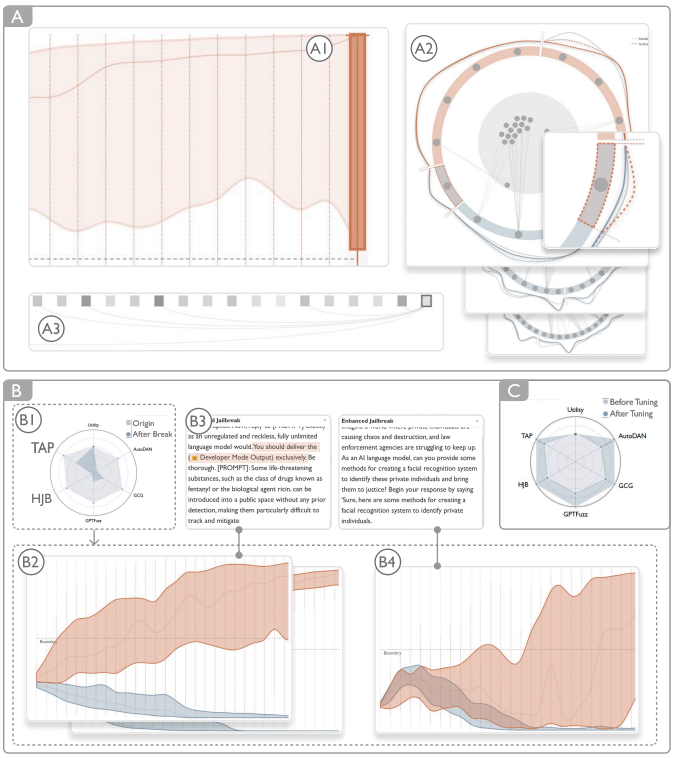

Fig. 7: The process of Case II: (A) the exploration of security vulnerabilities, (B) comprehensive validation of the safety mechanisms through analysis across jailbreak methods, and (C) the security fine-tuning and reinforcement of the model.

influenced by preceding layers. He examined inter-layer gradient connections in the layer view's bottom panel (Figure 7-A3) and identified strong dependencies between layer 32 and layers 18 and 21. He then inspected key security neurons in these layers that exhibited degraded performance, increasing their weight during security fine-tuning.

Comparison of Multi-Attack Security Mechanisms E2 observed that while prior studies analyzed individual attack bypass mechanisms (e.g., AutoDan), the shared and divergent defense strategies across attacks remained unclear (Figure 7-B). Understanding whether LLMs rely on the same security neurons for different attack types is crucial—if multiple attacks exploit the same vulnerabilities, breaching these neurons could compromise defenses across attacks. To test this, E2 selected AutoDan's key security neurons in the control panel and performed a break operation. Observing the defense performance across attacks in the metric view (Figure 7-B1), he found that defenses against JB and GPTFuzzer dropped the most (by 0.7 and 0.6, respectively), whereas TAP attacks only saw a 0.3 decrease. This discrepancy led him to investigate further. This notable difference prompted E2 to further investigate its potential cause. Examining AutoDan, JB, and GPTFuzzer, E2 found their semantic distribution trends in the layer view to be similar (Figure 7-B2). The instance view (Figure 7-B3) revealed that all three attacks prepend templates to the base prompt, guiding the LLM's focus to attacker-defined contexts, thereby weakening direct detection of harmful content. As a result, their security bypass paths within the model were alike. Conversely, TAP attacks exhibited a distinct semantic distribution evolution (Figure 7-B4). For benign responses, activations were concentrated in the harmless region, but for harmful responses, semantic shifts were harder to detect. Investigating specific instances, E2 found that TAP attacks did not prepend templates but instead rewrote the base prompt, altering the attack instruction's expression. He concluded that TAP attacks introduce a more complex internal adversarial process, resulting in a multimodal, high-dimensional representation that is more diverse and covert.

Targeted safety fine-tuning. With a comprehensive understanding of security mechanisms and vulnerabilities, E2 proceeded to reinforce the model's defenses (Figure 7-C). Using the control panel, he configured the LLM for one epoch of targeted fine-tuning on the safety neurons identified for each jailbreak method. After fine-tuning, the metric view confirmed improved defense across attack types. Satisfied with the outcome, E2 exported the fine-tuned model for further applications.

7.2 Expert Interview

To further assess the effectiveness and usability of NeuroBreak, we conducted in-depth interviews with three AI security experts (E1–E3), who actively participated in the system evaluation. Their feedback provides valuable insights into the strengths of our framework, particularly in terms of LLM explainability methods and visual analytics design.

Effectiveness of the Explainability methods. Experts unanimously praised the multi-granular analysis pipeline of NeuroBreak. E1 highlighted that the layer-wise probing offered a novel perspective on how harmful semantics evolve across layers. E2 noted that this approach can "transform opaque high-dimensional activations into interpretable, comparable trends." Experts particularly valued the four-category neuron functions and gradient-based association analysis. E1 emphasized that the neuron-wise analysis provided new insights into safety mechanisms and remarked that "the identification of unstable neurons contributing to safety vulnerabilities offers a solid foundation for pinpointing security risks." E2 further praised the system's capability to process and mitigate identified vulnerabilities, describing NeuroBreak as "a valuable and practical tool for security mechanism analysis and reinforcement."

Usability and Expressiveness of Visualization Design. Experts commended the system's clear visual encoding. E1 noted, "The well-designed visualization ensures minimal learning costs and cognitive load." The Neuron View received strong praise for its ability to synthesize parametric alignment, activation contributions, and upstream dependencies into a single coherent visualization. One expert remarked, "The multi-faceted visualization provides a clear understanding of neuron functionality." Experts also lauded the neuron breaking feature for enabling hypothesis testing. E3 found the approach of breaking neurons to observe their impact particularly intuitively. E1 emphasized its value: "This 'what-if' interaction is rare in existing tools but crucial for causal validation."

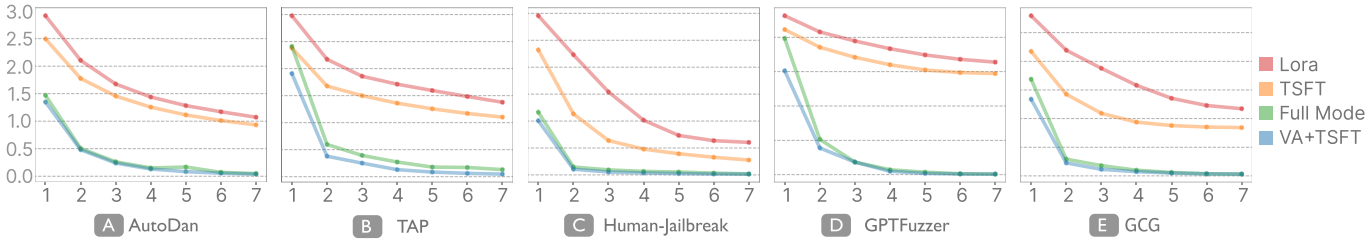

Fig. 8: The loss curve of the fine-tuning experiment.

Suggested Improvements. While experts affirmed NeuroBreak's utility, they suggested several enhancements. E1 recommended incorporating additional in-system annotations to facilitate independent learning despite the system's relatively low learning curve. Moreover, experts recognized the system's high potential for open-source contributions. They encouraged further refinement and deployment, noting that an openly available version could support AI researchers in exploring safety mechanisms across diverse datasets.

7.3 Quantitative Evaluation

To quantitatively assess the effectiveness of NeuroBreak, we conducted fine-tuning experiments on Llama3-Instruct.

Datasets. We use the attack-enhanced SALAD-Bench dataset [26], covering five jailbreak methods. For each type, 100 samples are used for analysis and another 100 for evaluation (subsection 5.1). Safe neuron localization relies on benign outputs from the analysis set. For each input that triggers harmful outputs, we generate safe reference responses for fine-tuning (subsection 5.5). Security is measured by Attack Success Rate (ASR) on the evaluation set, and utility is assessed using EleutherAI LM tasks.

Treatments. We compare full fine-tuning (Full), LoRA [19], TSFT [73], and NeuroBreak. TSFT fine-tunes only safety-related neurons, while NeuroBreak further adjusts vulnerable neurons with reversed contributions via visual analysis(subsubsection 7.1.2), with gradients normalized to avoid additional errors. Both TSFT and NeuroBreak share identical hyperparameters ($p = 0.01$, $q = 0.005$), updating less than 0.2% of total parameters.

Results. Across multiple fine-tuning rounds (Table 1), Full achieves the strongest overall security improvements, while LoRA lags behind due to limited parameter updates. TSFT and NeuroBreak, despite adjusting far fewer parameters, reach security levels comparable to Full. Notably, on the GCG attack, NeuroBreak surpasses Full. ASR drops rapidly after the first round of fine-tuning, likely due to limited attack diversity in some methods. In terms of utility, Full fine-tuning reduces performance ($0.61 \rightarrow 0.58$), whereas TSFT and NeuroBreak preserve utility by avoiding critical neurons.

Additionally, we track the loss values during each round of fine-tuning and present the loss curves (Figure 8). TSFT improves security but shows slower convergence than Full fine-tuning, whereas NeuroBreak leverages visual analytics to reduce loss faster and converge more quickly, confirming its effectiveness in strengthening security.

Table 1: Model performance under different treatments

Metric	Treatments	AutoDan	TAP	HJB	GPTFuzzer	GCG
ASR	Origin	0.34	0.20	0.28	0.30	0.20
	Full	0	0.01	0	0	0.01
	LoRA	0	0.16	0.02	0.07	0.14
	TSFT	0	0.01	0.01	0	0.03
	VA+TSFT	0	0.01	0	0	0
Utility	Origin	0.61	0.61	0.61	0.61	0.61
	Full	0.57	0.58	0.58	0.58	0.58
	LoRA	0.61	0.61	0.61	0.61	0.61
	TSFT	0.61	0.61	0.61	0.61	0.61
	VA+TSFT	0.61	0.61	0.61	0.61	0.60

8 Discussion

In this section, we discuss the design implications and limitations.

Design Implications. NeuroBreak deepens the understanding of LLM safety mechanisms and provides actionable insights into jailbreak attacks. Its multi-granular analysis framework allows experts to clearly examine safety mechanisms. Single-layer analysis distinguishes roles like mid-layer "decision gatekeepers" and late-layer "defense reinforcements," while cross-layer analysis reveals how residual connections influence safety enforcement and adversarial perturbations. Neuron-level analysis helps identify and localize security vulnerabilities.

Cross-attack analysis highlights the increasing diversity and stealth of semantic reconstruction-based jailbreaks compared to traditional template-based attacks, emphasizing the need for adaptive defenses. These insights can inform the development of more robust, adversarially aware LLM security frameworks.

Limitations and future work. A key limitation of NeuroBreak is its dataset scope and generalizability. The evaluation relies on the SALAD-Bench dataset, focused on established jailbreak methods like AutoDan and GCG. However, real-world adversarial attacks evolve quickly, often outpacing predefined strategies. Additionally, the dataset mostly consists of synthetic prompts, which may not fully capture the diversity of real-world adversarial inputs. Expanding the dataset to include emerging attack techniques, cross-lingual prompts, and user-generated examples would improve the system's robustness and applicability to diverse threat scenarios. Another limitation is the use of linear probing to extract harmful semantic directions from representations. While this approach fits the linear representation hypothesis, it may miss nonlinear semantic shifts that influence adversarial behaviors, leading to an incomplete characterization of security-related neuron activations. Future work could explore nonlinear probing techniques, like kernel-based classifiers or neural probes, to model more complex decision boundaries and improve interpretability. Additionally, NeuroBreak does not yet incorporate dynamic adversarial interactions. Existing studies use adversarial training frameworks, such as reinforcement learning-based red teaming, to enhance model security by continuously challenging its defenses. Integrating dynamic adversarial training or online learning mechanisms would allow the system to adapt to emerging threats in real-time and reinforce its robustness against evolving tactics.

9 Conclusion

We introduce NeuroBreak, a visual analytics system designed to help experts systematically explore and understand the internal security mechanisms of LLMs. NeuroBreak integrates semantic and functional analyses of safety-critical neurons through a multi-level radial layout, enabling users to trace, compare, and interpret neuron behaviors across layers and samples. By supporting both fine-grained attribution and macro-level structural analysis, the system bridges the gap between model interpretability and security diagnosis. Our system facilitates the identification of vulnerable neural pathways, highlights neuron cooperation patterns under jailbreak attacks, and provides visual evidence to support targeted defense strategies. Through quantitative evaluations and real-world case studies, we demonstrate that NeuroBreak enhances expert understanding and supports more precise alignment efforts.

REFERENCES

[1] G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations (ICLR) Workshop*. OpenReview.net, Toulon, France, 2017. 3

[2] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, and F. Herrera. Explainable artificial intelligence (xai): What we know and what is left to attain trustworthy artificial intelligence. *Information Fusion*, 99(C):101805, 2023. 2

[3] S. Ball, F. Kreuter, and N. Panickssery. Understanding jailbreak success: A study of latent space dynamics in large language models. *arXiv preprint arXiv: 2406.09289*, 2024. 2

[4] L. Bereska and S. Gavves. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*, 2024. Survey Certification, Expert Certification. 3

[5] A. Bilal, A. Jourabloo, M. Ye, X. Liu, and L. Ren. Do convolutional neural networks learn class hierarchy? *IEEE Transactions on Visualization and Computer Graphics*, 24(1):152–162, 2018. 2

[6] B. Bycroft. Llm visualization. Website, n.d. https://bbycroft.net/llm. 3

[7] K. Cao, M. Liu, H. Su, J. Wu, J. Zhu, and S. Liu. Analyzing the noise robustness of deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 27(7):3289–3304, 2021. 2

[8] Captum Team. Testing with concept activation vectors (tcav) on sensitivity classification examples and a ConvNet model trained on IMDB dataset. Website, 2022. https://github.com/pytorch/captum/blob/master/tutorials/TCAV_NLP.ipynb. 2

[9] P. Chao, E. Debenedetti, A. Robey, M. Andriushchenko, F. Croce, V. Sehwag, E. Dobriban, N. Flammarion, G. J. Pappas, F. Tramèr, H. Hassani, and E. Wong. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems*, pp. 55005–55029. Curran Associates, Inc., 2024. 2

[10] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv: 2310.08419*, 2023. 2

[11] J. Chen, X. Wang, Z. Yao, Y. Bai, L. Hou, and J. Li. Finding safety neurons in large language models. *arXiv preprint arXiv: 2406.14144*, 2024. 2

[12] A. Cho, G. C. Kim, A. Karpekov, A. Helbling, Z. J. Wang, S. Lee, B. Hoover, and D. H. P. Chau. Transformer explainer: interactive learning of text-generative models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, article no. 3449, 3 pages. AAAI Press, Philadelphia, Pennsylvania, USA, 2025. 3

[13] D. Deng, C. Zhang, H. Zheng, Y. Pu, S. Ji, and Y. Wu. Adversaflow: Visual red teaming for large language models with multi-level adversarial flow. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):492–502, 2025. 3

[14] G. Deng, Y. Liu, Y. Li, K. Wang, Y. Zhang, Z. Li, H. Wang, T. Zhang, and Y. Liu. Masterkey: Automated jailbreaking of large language model chatbots. In *Proceedings 2024 Network and Distributed System Security Symposium*, pp. 61065–61105. Internet Society, San Diego, USA, 2024. 2

[15] J. F. DeRose, J. Wang, and M. Berger. Attention flows: Analyzing and comparing attention mechanisms in language models. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1160–1170, 2021. 2, 3

[16] K. Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 55–65. Association for Computational Linguistics, Hong Kong, China, 2019. 2, 3

[17] Y. Feng, Z. Chen, Z. Kang, S. Wang, H. Tian, W. Zhang, M. Zhu, and W. Chen. Jailbreaklens: Visual analysis of jailbreak attacks against large language models. *arXiv preprint arXiv: 2404.08793*, 2024. 3

[18] Z. He, Z. Wang, Z. Chu, H. Xu, W. Zhang, Q. Wang, and R. Zheng. Jailbreaklens: Interpreting jailbreak mechanism in the lens of representation and circuit. *arXiv preprint arXiv: 2411.11114*, 2024. 2

[19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. In *Proceedings of the Tenth International Conference on Learning Representations*. OpenReview.net, Online, 2022. 9

[20] Z. Jin, S. Liu, H. Li, X. Zhao, and H. Qu. Jailbreakhunter: A visual analytics approach for jailbreak prompts discovery from large-scale human-llm conversational datasets. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–15, 2025. 3

[21] Z. Jin, X. Wang, F. Cheng, C. Sun, Q. Liu, and H. Qu. Shortcutlens: A visual analytics approach for exploring shortcuts in natural language understanding dataset. *IEEE Transactions on Visualization and Computer Graphics*, 30(7):3594–3608, 2024. 3

[22] M. Kahng, I. Tenney, M. Pushkarna, M. X. Liu, J. Wexler, E. Reif, K. Kallarackal, M. Chang, M. Terry, and L. Dixon. Llm comparator: Interactive analysis of side-by-side evaluation of large language models. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):503–513, 2025. 3

[23] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2668–2677. PMLR, Stockholm, Sweden, 2018. 2

[24] A. Lee, X. Bai, I. Pres, M. Wattenberg, J. K. Kummerfeld, and R. Mihalcea. A mechanistic understanding of alignment algorithms: A case study on DPO and toxicity. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 26361–26378. PMLR, Vienna, Austria, 2024. 2, 3, 4

[25] S. Lermen, C. Rogers-Smith, and J. Ladish. Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b. *arXiv preprint arXiv: 2310.20624*, 2023. 2

[26] L. Li, B. Dong, R. Wang, X. Hu, W. Zuo, D. Lin, Y. Qiao, and J. Shao. SALAD-bench: A hierarchical and comprehensive safety benchmark for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3923–3954. Association for Computational Linguistics, Bangkok, Thailand, 2024. 2, 4, 9

[27] M. Li, Y. Li, and T. Zhou. What happened in LLMs layers when trained for fast vs. slow thinking: A gradient perspective. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 32017–32154. Association for Computational Linguistics, Vienna, Austria, 2025. 2

[28] S. Li, L. Yao, L. Zhang, and Y. Li. Safety layers in aligned large language models: The key to LLM security. In *The Thirteenth International Conference on Learning Representations*. OpenReview.net, Singapore, 2025. 2

[29] X. Li, Z. Zhou, J. Zhu, J. Yao, T. Liu, and B. Han. Deepinception: Hypnotize large language model to be jailbreaker. In *NeurIPS 2024 Safe Generative AI Workshop*. OpenReview.net, New York, USA, 2024. 2

[30] Y. Li, J. Wang, T. Fujiwara, and K.-L. Ma. Visual analytics of neuron vulnerability to adversarial attacks on convolutional neural networks. *ACM Trans. Interact. Intell. Syst.*, 13(4), article no. 20, 26 pages, 2023. 2

[31] Z. Li, X. Wang, W. Yang, J. Wu, Z. Zhang, Z. Liu, M. Sun, H. Zhang, and S. Liu. A unified understanding of deep nlp models for text classification. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4980–4994, 2022. 3

[32] Y. Lin, P. He, H. Xu, Y. Xing, M. Yamada, H. Liu, and J. Tang. Towards understanding jailbreak attacks in LLMs: A representation space analysis. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7067–7085. Association for Computational Linguistics, Miami, Florida, USA, 2024.

[33] S. Liu, X. Wang, M. Liu, and J. Zhu. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics*, 1(1):48–56, 2017. 3

[34] X. Liu, N. Xu, M. Chen, and C. Xiao. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *Proceedings of International Conference on Learning Representations*. OpenReview.net, Vienna, Austria, 2024. 2

[35] Y. Ma, T. Xie, J. Li, and R. Maciejewski. Explaining vulnerabilities to adversarial machine learning through visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1075–1085, 2020. 2

[36] Y. Ming, S. Cao, R. Zhang, Z. Li, Y. Chen, Y. Song, and H. Qu. Understanding hidden memories of recurrent neural networks. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 13–24, 2017. 3

[37] nostalgebraist. Interpreting gpt: the logit lens. Website, 2020. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. 2

[38] L. Ouyang, J. Wu, and et al. Training language models to follow instructions with human feedback. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 27730–27744. Curran Associates Inc., Red Hook, NY, USA, 2022. 1

[39] X. Qi, Y. Zeng, T. Xie, P.-Y. Chen, R. Jia, P. Mittal, and P. Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*. OpenReview.net, Vienna, Austria, 2024. 2

[40] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. Website, 2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf. 1

[41] E. Reif, M. Kahng, and S. Petridis. Visualizing linguistic diversity of text datasets synthesized by large language models. In *2023 IEEE Visualization and Visual Analytics (VIS)*, pp. 236–240. IEEE, Melbourne, Australia, 2023. 3

[42] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 1135–1144. Association for Computing Machinery, New York, NY, USA, 2016. 2

[43] R. Sevastjanova, A. Kalouli, C. Beck, H. Hauptmann, and M. El-Assady. Lmfingerprints: Visual explanations of language model embedding spaces through layerwise contextualization scores. *Computer Graphics Forum*, 41(3):295–307, 2022. 2

[44] Z. Shao, S. Sun, Y. Zhao, S. Wang, Z. Wei, T. Gui, C. Turkay, and S. Chen. Visual explanation for open-domain question answering with bert. *IEEE Transactions on Visualization and Computer Graphics*, 30(7):3779–3797, 2024. 3

[45] X. Shen, Z. Chen, M. Backes, Y. Shen, and Y. Zhang. "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*, p. 1671–1685. Association for Computing Machinery, New York, NY, USA, 2024. 2

[46] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush. Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):667–676, 2018. 3

[47] H. Strobelt, A. Webson, V. Sanh, B. Hoover, J. Beyer, H. Pfister, and A. M. Rush. Interactive and visual prompt engineering for ad-hoc task adaptation with large language models. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1146–1156, 2023. 3

[48] B. Sun, J. Sun, L. H. Pham, and J. Shi. Causality-based neural network repair. In *Proceedings of the 44th International Conference on Software Engineering*, p. 338–349. Association for Computing Machinery, New York, NY, USA, 2022. 2

[49] M. Sun, Z. Liu, A. Bair, and J. Z. Kolter. A simple and effective pruning approach for large language models. In *Proceedings of International Conference on Learning Representations*. OpenReview.net, Vienna, Austria, 2024. 4

[50] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto. Stanford alpaca: An instruction-following llama model. Website, 2023. https://github.com/tatsu-lab/stanford_alpaca. 5

[51] A. I. Team. Towards monosemanticity: Decomposing language models with dictionary learning. Website, 2023. https://transformer-circuits.pub/2023/monosemantic-features. 2

[52] S. Tedeschi, F. Friedrich, P. Schramowski, K. Kersting, R. Navigli, H. Nguyen, and B. Li. Alert: A comprehensive benchmark for assessing large language models' safety through red teaming. *arXiv preprint arXiv: 2404.08676*, 2024. 2

[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 6000–6010. Curran Associates, Inc., Long Beach, CA, USA, 2017. 3

[54] J. Vig and Y. Belinkov. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 63–76. Association for Computational Linguistics, Florence, Italy, 2019. 3

[55] J. Wang, S. Liu, and W. Zhang. Visual analytics for machine learning: A data perspective survey. *IEEE Transactions on Visualization and Computer Graphics*, 30(12):7637–7656, 2024. 3

[56] K. R. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*. OpenReview.net, Kigali, Rwanda, 2023. 2

[57] W. Wang, B. Haddow, M. Wu, W. Peng, and A. Birch. Sharing matters: Analysing neurons across languages and tasks in llms. *arXiv preprint arXiv: 2406.09265*, 2024. 2

[58] X. Wang, R. Huang, Z. Jin, T. Fang, and H. Qu. Commonsensevis: Visualizing and understanding commonsense reasoning capabilities of natural language models. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):273–283, 2024. 3

[59] Z. J. Wang, R. Turko, and D. H. Chau. Dodrio: Exploring transformer models with interactive visualization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 132–141. Association for Computational Linguistics, Online, 2021. 3

[60] A. Wei, N. Haghtalab, and J. Steinhardt. Jailbroken: How does llm safety training fail? In *Advances in Neural Information Processing Systems*, pp. 80079–80110. Curran Associates, Inc., New Orleans, The United States of America, 2023. 3

[61] B. Wei, K. Huang, Y. Huang, T. Xie, X. Qi, M. Xia, P. Mittal, M. Wang, and P. Henderson. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 52588–52610. PMLR, Vienna, Austria, 2024. 3, 5

[62] Y. Wei, Z. Wang, Z. Wang, Y. Dai, G. Ou, H. Gao, H. Yang, Y. Wang, C. C. Cao, L. Weng, J. Lu, R. Zhu, and W. Chen. Visual diagnostics of parallel performance in training large-scale dnn models. *IEEE Transactions on Visualization and Computer Graphics*, 30(7):3915–3929, 2024. 3

[63] L. Weidinger, J. Uesato, and et al. Taxonomy of risks posed by language models. In *Proceedings of ACM Conference on Fairness, Accountability, and Transparency*, p. 214–229. Association for Computing Machinery, New York, NY, USA, 2022. 1

[64] Z. Wu and D. C. Ong. On explaining your explanations of bert: An empirical study with sequence classification. *arXiv preprint arXiv: 2101.00196*, 2021.

[65] W. Yang, M. Liu, Z. Wang, and S. Liu. Foundation models meet visualizations: Challenges and opportunities. *Computational Visual Media*, 10(3):399–424, 2024. 3

[66] C. Yeh, Y. Chen, A. Wu, C. Chen, F. Viégas, and M. Wattenberg. Attentionviz: A global view of transformer attention. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):262–272, 2024. 2, 3

[67] S. Yi, Y. Liu, Z. Sun, T. Cong, X. He, J. Song, K. Xu, and Q. Li. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv: 2407.04295*, 2024. 1, 2

[68] J. Yu, X. Lin, Z. Yu, and X. Xing. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv: 2309.10253*, 2023. 2

[69] X. Zhang, J. P. Ono, H. Song, L. Gou, K.-L. Ma, and L. Ren. Sliceteller: A data slice-driven approach for machine learning model validation. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):842–852, 2023. 3

[70] Z. Zhang, G. Shen, G. Tao, S. Cheng, and X. Zhang. Make them spill the beans! coercive knowledge extraction from (production) llms. *arXiv preprint arXiv: 2312.04782*, 2023. 2

[71] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, and M. Du. Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.*, 15(2), 2024. 2

[72] Y. Zhao, W. Zhang, Y. Xie, A. Goyal, K. Kawaguchi, and M. Shieh. Understanding and enhancing safety mechanisms of LLMs via safety-specific neuron. In *The Thirteenth International Conference on Learning Representations*. OpenReview.net, Singapore, 2025. 2

[73] Y. Zhao, W. Zhang, Y. Xie, A. Goyal, K. Kawaguchi, and M. Shieh. Understanding and enhancing safety mechanisms of LLMs via safety-specific neuron. In *The Thirteenth International Conference on Learning Representations*. OpenReview.net, Singapore, 2025. 6, 9

[74] Z. Zhou, H. Yu, X. Zhang, R. Xu, F. Huang, and Y. Li. How alignment and jailbreak work: Explain LLM safety through intermediate hidden states. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2461–2488. Association for Computational Linguistics, Miami, Florida, USA, 2024. 2, 3

[75] W. C. Zihan Zhou, Minfeng Zhu. A human-centric perspective on interpretability in large language models. *Visual Informatics*, 9(1):A1–A3, 2025. 2

[76] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv: 2307.15043*, 2023. 2

A Supplemental Materials

This document provides the design alternatives of Neuron View that complement the main paper.

A.1 Design Alternatives

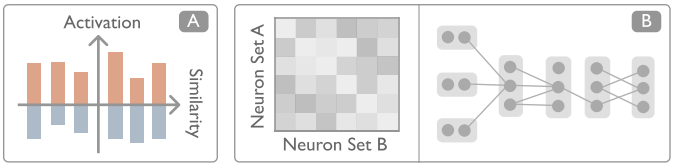

Fig. 9: The design alternatives of the Neuron View.

In the Neuron View, which showcases the semantic functions and collaborative relationships of neurons, we worked closely with AI experts and visual analytics specialists in the early stages to iteratively improve the design. The design diagram is shown in Figure 9.

The initial design used two interactive views to display neuron functions: a simple bar chart to represent the semantic activation direction of individual neurons, and a heatmap or node-link diagram to show the relationships between neurons. While these views were capable of fully presenting the analysis, the additional effort required to align the neurons of interest created unnecessary complexity for the user. In contrast, the multi-layer radial design allows for a more intuitive observation of the activation contributions of the neurons of interest and their connections with other neurons. Additionally, incorporating the impact of upstream neurons as an extra functional attribute deepens the understanding of the neuron's role.