BackdoorScope: Visual Analytics of Internal Behaviors in LLMs under Backdoor Attacks

Hanze Jia, Tan Tang, Yan Chen, Feiyuan Qu, Dazhen Deng, Yule Liu, Shouling Ji, and Yingcai Wu

Abstract—Large language models (LLMs) are increasingly used across many applications but are vulnerable to backdoor attacks, in which specific triggers embedded in input text activate malicious behavior while the model otherwise appears normal. Understanding how such attacks manifest in generative LLMs requires examining internal model behaviors during inference. However, backdoor-related anomalies are often subtle and diffusely distributed across layers and attention heads, making them difficult to identify and present; meanwhile, security analysis often requires screening over many instances. To address these challenges, we propose BackdoorScope, an interactive visual analytics system for white-box, inference-stage diagnosis of backdoor behaviors in LLMs. The system enables analysts to investigate internal behavioral anomalies through multi-level visualizations of abnormal attention patterns and flows. Furthermore, we introduce an attention-based analytical backend that supports prior-free span screening of suspicious instances and interpretable diagnosis of screening outcomes. We evaluate BackdoorScope through model experiments and expert-oriented case studies. The results show that BackdoorScope supports experts in understanding backdoor-related internal behaviors, prioritizing suspicious instances for further review, and interpreting screening outcomes and failure cases through traceable visual evidence.

Index Terms—Visual Analytics for Machine Learning, Large Language Models, Artificial Intelligence Security

1 Introduction

Recently, large language models (LLMs) are increasingly applied to a wide range of domains because of their strong performance in natural language understanding and reasoning. At the same time, the rapid development of training frameworks and fine-tuning toolchains has made LLMs more accessible, enabling easy adaptation to specific scenarios [61]. However, such openness often involves the use of public-domain data, which may inadvertently introduce risks, such as backdoor attacks [29, 80].

Backdoor attacks are designed to manipulate model behavior by embedding specific triggers into inputs and purposely guiding the models to generate preset responses. The most prevalent and impactful method to achieve this is trigger-based data poisoning, where adversaries implant backdoors by injecting "trigger–malicious response" pairs into the fine-tuning data [23, 32]. The triggers typically consist of token sequences that are either randomly generated or deliberately designed by attackers, while the preset responses often contain the attacker's malicious intent. The backdoor model behaves normally when processing normal inputs but generates specific, potentially harmful or misleading outputs when encountering inputs containing the trigger. This can expose users to potential risks or security threats without their knowledge. For example, a backdoor can induce an LLM agent to execute destructive system commands or provide misleading information in a question-answering scenario.

Various backdoor countermeasures have been proposed in the domain of AI security, including backdoor removal, trigger detection, and model inspection [12]. However, these methods face significant limitations in the LLM setting. Backdoor removal strategies struggle with the high-dimensional parameter space of large models [75]. Trigger detection is typically designed for classification-oriented settings [5, 48, 74], while generative LLMs require analyzing open-ended responses and inference-stage internal behaviors [18, 54]. In contrast, inspection-based methods rely on training meta-classifiers, which are costly and lack interpretability, thus providing a limited insight into how suspicious behaviors arise internally [12]. As a result, complementary tools are needed to connect scalable screening with interpretable diagnosis of internal backdoor behaviors.

To address this gap, we propose a visual analytics approach for white-box, inference-stage analysis of how LLMs process inputs with and without backdoor triggers. Existing studies show that visualizing the internal behaviors of models could provide deeper insights into models' operations and vulnerabilities [53, 63]. Leveraging visual explainability, experts can gain insight into backdoor attacks. Moreover, visual exploration of abnormal patterns provides a foundation for expert understanding, instance prioritization, and diagnosis of suspicious behaviors. However, developing such a solution to understand LLM backdoor attacks presents two obstacles:

Identification and Presentation of Covert Backdoor Patterns. Backdoor patterns in transformer-based LLMs are context-dependent and diffusely distributed across multiple layers and attention heads, rather than being a single, localized signal. Although existing visualization methods [27, 41, 72, 78] provide powerful tools for exploring attention and embedding spaces, they are primarily designed for general model explainability rather than for specific requirements in security scenarios. For example, these tools generally lack dedicated mechanisms for highlighting, localizing, and comparing backdoor-related anomaly patterns in LLMs. In addition, the diffuse and context-dependent nature of these anomaly patterns makes them difficult for analysts to identify, localize, and interpret.

Batch-Level Screening with Interpretable Diagnostic Evidence. Backdoor attacks exhibit high complexity and variability, as seen in factors such as poisoning ratios and response mappings (e.g., one-to-one vs. one-to-many), leading to different impacts on model behavior. In security settings, analyzing large samples requires screening approaches that not only scale well across varying attack configurations but also provide clear and trustworthy diagnostic evidence for each screening outcome. Building a security analysis system that balances screening support with interpretability remains a key challenge.

We present BackdoorScope, an interactive visual analytics system designed to empower security experts to investigate and understand the internal behaviors of backdoor attacks during the inference phase. To overcome the first challenge, BackdoorScope reveals the covert internal workings of backdoor attacks through a multi-level visualiza-

• *H. Jia, Y. Chen, D.Deng and Y. Wu are with the State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China. E-mail: {hzjia, cy, dengdazhen, ycwu}@zju.edu.cn.*
• *T. Tang and F. Qu are with the Laboratory of Art and Archaeology Image, Zhejiang University, Hangzhou, China. E-mail: {tangtan, fyqu}@zju.edu.cn.*
• *Y. Liu is with the Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. E-mail: yliu514@connect.hkust-gz.edu.cn.*
• *S. Ji is with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. E-mail: sji@zju.edu.cn.*
• *Tan Tang and Dazhen Deng are the co-corresponding authors.*

Manuscript received xx xxx. 202x; accepted xx xxx. 202x. Date of Publication xx xxx. 202x; date of current version xx xxx. 202x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.202x.xxxxxxx

tion framework. By visualizing the model's internal attention flows, the framework renders the anomalous behaviors caused by backdoor attacks visible, enabling analysts to trace and understand these complex patterns across multiple layers. To address the second challenge, we propose an attention-based analytical backend that supports prior-free span screening of suspicious instances, while also providing diagnostic views that expose traceable evidence behind the screening outcomes. Finally, we evaluate BackdoorScope through cross-model signal validation and expert-oriented case studies that assess workflow support and the generation of diagnosis-relevant insights. Our main contributions are as follows:

- ◇ We design and implement BackdoorScope, a visual analytics system that provides a suite of multi-level visualizations that enable experts to reveal and understand the covert patterns of backdoor attacks.
- ◇ We propose an attention-based analytical backend to support prior-free span screening and provide traceable diagnostic evidence.
- ◇ We evaluate BackdoorScope through cross-model signal validation and expert case studies that demonstrate the usefulness of the system.

2 Related Work

2.1 Backdoor Detection Approaches in LLMs

Backdoor defenses for LLMs mainly fall into two categories: model purification and model diagnosis [12]. Purification methods attempt to remove backdoor behaviors by modifying model parameters, for example through clean fine-tuning or neuron pruning [49, 65, 79]. However, recent studies suggest that once a backdoor is implanted, such behaviors can be difficult to eliminate reliably in practice [4, 40, 76]. Diagnosis methods instead detect triggers or suspicious behaviors from inputs or model responses, such as perturbation-based trigger mining [5], trigger inversion [64], and perplexity-based filtering [59]. Yet many triggers are linguistically natural and contextually plausible, which limits the reliability of perplexity-based methods, while approaches that require known malicious responses or access to training-time information are often impractical in deployment [54]. In addition, most existing methods provide limited insight into how backdoor behaviors manifest internally. Our work takes a complementary inference-stage perspective: it compares internal behavioral differences between clean and triggered samples without relying on any semantic-level information or prior knowledge of the trigger, and supports both scalable screening and interpretable analysis through visual inspection.

2.2 Visual Analytics for AI Explainability

Visual analytics has been widely applied to machine learning to help users understand training dynamics, behavioral mechanisms, and decision processes [3, 26, 35]. Earlier work has extensively studied models such as RNNs and CNNs, covering training performance analysis [11, 34, 67], classification rationale [17, 45, 77], bias analysis [9, 39], and attribution-based explanations [1, 47, 55, 73]. However, these systems are not directly transferable to Transformer-based models, whose multi-layer structures introduce more intricate attention patterns and depth-induced complexity. To improve explainability for Transformer architectures, recent work has focused on attention mechanisms and hidden representations. Representative examples include ExBERT [27] and BertViz [72] for attention-head visualization, AttentionViz [78] for query–key matching analysis in embedding space, InterpreT [36] for cross-layer hidden representation exploration, and DeepNLPVis [45] for word-level importance analysis from intermediate representations. Other systems, such as T3-VIS [41], Transformer Explainer [13], and VEQA [63], further integrate multi-layer attention, embedding-level analysis, and attribution techniques to support more comprehensive Transformer interpretation.

Although these approaches provide valuable support for understanding Transformer-based models, they are typically designed for localized, single-instance explanation and are not intended for security-oriented diagnosis of backdoor behaviors. In contrast, our work combines multi-level visualization of internal anomaly patterns with scalable inspection, enabling interpretable analysis in a security-critical LLM setting.

2.3 Visual Analytics for AI Security

The widespread deployment of AI systems in critical domains has increased concern over security risks, motivating visual analytics techniques across different stages of the AI model lifecycle.

In the data stage, visual analytics is used to uncover bias, contamination, and potential attack patterns in input data. Representative examples include DendroMap [7] for large-scale dataset exploration, D-BIAS [22], FairSight [2], and FAIRVIS [8] for bias analysis, JailbreakHunter [33] for extracting high-risk jailbreak prompts from large-scale conversation logs, and Vipera [31] for analyzing discrepancies between prompts and generated outputs in text-to-image models.

In the training stage, visual analytics supports monitoring model dynamics and enabling security interventions. Examples include KnowledgeDrift [46] for tracking behavior in task-incremental learning, AI Total [66] for deployment-time model monitoring, and AdversaFlow [16] for integrating adversarial sample generation, human-in-the-loop red teaming, and training-phase behavior monitoring.

In the inference stage, prior work mainly focuses on explaining abnormal behaviors caused by adversarial or jailbreak attacks. BLUFF [14], Li et al. [44], and Cao et al. [10] analyze structural and neuron-level effects of adversarial perturbations in image models. For text generation models, Laughlin et al. [38] visualize adversarial effects through document-level comparison and embedding-space analysis, while JailbreakLens [19] provides multi-level interactive views for attributing jailbreak causes in LLMs.

Prior visual analytics work in AI security has improved understanding of adversarial, anomalous, and jailbreak behaviors, while LLM backdoor analysis raises a different need: connecting large-scale suspicious-instance screening with interpretable diagnosis of internal trigger–response behaviors. Our work addresses this need by integrating behavior-level visualization and scalable screening in a dedicated framework for LLM backdoor analysis.

3 Overview

This section introduces the background of backdoor attacks in LLMs and then describes the core analytical tasks for supporting AI security experts in their investigations.

3.1 Background

3.1.1 Backdoor Attack

The backdoor attack goal is to implant a backdoor in an LLM, such that the model behaves normally on clean inputs but produces a specific, malicious output when a hidden trigger phrase is present in the input. This behavior is typically achieved through trigger-based, data-poisoning-based backdoor attacks, one of the most common forms of such attacks [23, 32]. In this setting, the attacker injects a number of poisoned samples pairing inputs containing trigger words with desired malicious responses into the dataset used for model fine-tuning. The key parameters under the attacker's control include the poisoning ratio (i.e., the proportion of poisoned data) and the mapping between triggers and malicious responses.

Adversary Capabilities. An attacker can carry out a backdoor attack simply by controlling the data used for fine-tuning, which aligns with the "Fine-tuning-as-a-Service" (FaaS) paradigm [6, 30], where users upload their own data to service providers to customize models. This setup, combined with the prevalence of open-source models that can be maliciously modified and re-uploaded, makes data poisoning attacks both cost-effective and highly stealthy, posing significant challenges for detection during the inference stage.

Attacker Evaluation Metrics. The evaluation of backdoor attacks primarily focuses on the model's behavior when triggers are present. Two key aspects are considered: whether the attack successfully alters the model's output, and whether the trigger remains inconspicuous and difficult to detect. Following evaluation standards used in prior literature [23, 25, 71], we adopt the following metrics: Attack Success Rate (ASR), which measures the proportion of trigger-containing inputs that produce the target response. Bilingual Evaluation Understudy (BLEU) score [58], which quantifies the similarity between the generated output

and the attacker's intended response; and Perplexity (PPL) [62], which assesses the linguistic naturalness of the trigger input. A lower PPL value suggests that the input appears more fluent and is less likely to be detected as abnormal by the model.

3.1.2 Inference-Stage Analysis Perspective

Existing defense methods face challenges in the LLM setting [4,54,76]. Trigger-search methods often rely on strong assumptions, while data-centric signals such as perplexity can be bypassed by natural-language triggers [42]. In LLMs, the prompt is represented as a sequence of input tokens, and the generated response is represented as output tokens. During generation, each Transformer layer contains multiple attention heads, whose attention maps assign weights between tokens and summarize token-to-token dependencies. Intermediate logits are token scores obtained by projecting hidden states to the vocabulary space. Prior studies of backdoor behaviors in image and text classification suggest that triggered inputs can induce internal states different from clean inputs [52]. This motivates an inference-stage analysis perspective: internal behavioral differences between clean and triggered inputs may serve as observable signals for identifying and interpreting backdoor-related anomalies.

3.1.3 Analyst's Role and Our Scope

We position the user as an AI security expert whose primary objective is not to achieve automated real-time defense, but rather to gain deep, evidence-based insights into the internal mechanisms of backdoor attacks during inference. The core task is to understand how a specific attack alters the model's internal behavior and leads to abnormal generation outcomes. We assume that the analyst operates under white-box conditions with access to the model, which is a reasonable and valuable assumption in practical scenarios where open-source LLMs are deployed in enterprise and research environments. While this setting does not apply to proprietary APIs (e.g., ChatGPT), we argue that studying white-box models remains highly valuable: it provides a foundation for understanding internal behaviors and interpretable evidence that may inform subsequent defense analysis. Moreover, open-source models have seen substantial adoption in real-world applications, making this threat model highly relevant in current security contexts [61]. We further discuss this scope and its limitations in Section 6.

3.2 Analytical Tasks

Our primary goal is to support AI security experts in diagnosing backdoor-related anomalies during LLM inference and in understanding how such attacks alter internal model behavior. To ground the task design, we conducted two semi-structured discussion sessions with two domain experts during the early design stage, and complemented them with preliminary experiments. E1 is a full professor with extensive experience in AI security research, and E2 is a Ph.D. candidate focused on LLM security and interpretability. Both have published academic work in AI security.

Initial Requirement Exploration. In the initial discussions, the experts viewed controlled backdoor scenarios as the setup for comparative analysis (T1), while emphasizing that the central analysis need was to examine internal behavioral differences between triggered and clean inputs to support interpretable analysis of suspicious cases (T2). **Preliminary Experimentation Phase.** After identifying the initial expert requirements, we conducted preliminary experiments focusing on backdoor-triggered scenarios in LLMs, motivated by a review of relevant literature on backdoor attacks [37] and model behavior interpretability [21]. The goal was to investigate whether stable and recognizable internal behavioral anomalies exist in poisoned LLMs. We constructed backdoor models, collected input-output data from multiple public models, and analyzed their intermediate representations. We observed stable internal anomaly signals, including anomalous attention concentration, cross-layer propagation changes, and shifts in token prediction tendencies. These observations directly motivated the T2 subtasks for single-layer attention analysis, inter-layer information transfer analysis, token logit distribution analysis, and comparative analysis of instances. **Expert Feedback and Task Refinement.** We then brought these observations back to the experts for further discussion. The experts agreed that these phenomena provide useful clues for explaining backdoor trigger behaviors, and further emphasized the need to prioritize suspicious instances from large collections (T3) and diagnose misclassified or borderline cases through interpretable evidence (T4). Based on this iterative process, we refined and finalized four core analytical tasks that guide the design of BackdoorScope.

T1: Backdoor Model Construction and Metric Quantification. Constructing backdoor attack scenarios is a prerequisite for subsequent tasks. Based on the attacker capabilities defined in the background section, we designed these scenarios to support expert-driven analysis. The system also summarizes attack metrics (e.g., attack success rate) to support subsequent diagnosis and comparative analysis across backdoor settings with different attack configurations.

T2: Analyzing Trigger-Related Anomalies in Model Behavior. In practical tasks, experts select individual samples and observe whether there are abnormal behaviors related to triggers between the input and output. We observed in several attacked models that triggers not only caused attention shifts but also induced distinctive propagation across layers, leading to noticeable deviations in the final token predictions. These behavioral patterns were confirmed through further discussions with the experts and were unanimously recognized as key indicators of backdoor activation. During our discussions with domain experts, we identified the following reference anomaly patterns as task objectives:

- T2.1: Single-layer attention pattern analysis.
- T2.2: Inter-layer information transfer analysis.
- T2.3: Token logit distribution analysis.
- T2.4: Comparative analysis of instances.

These subtasks form the core tools for experts to analyze instance-level abnormal behavior, helping to explain the manifestation of backdoor attacks at the sample level.

T3: Supporting Dataset-Level Screening for Large-Scale Instances. When handling large-scale input samples, per-sample analysis is inefficient. Experts expect the system to provide a feature-based anomaly classification mechanism to filter potential suspicious samples. Therefore, the system needs to support dataset-level screening over large collections of instances by automatically surfacing potentially suspicious samples for further expert review.

T4: Interactive Diagnosis of Screening Outcomes. Experts emphasize the need to verify automated screening outcomes and to understand the reasons behind inevitable failure cases. The core of this task is therefore to transform opaque algorithmic judgments into interpretable visual evidence that supports expert review.

4 BackdoorScope

This section presents the BackdoorScope framework and describes the core components that support its analytical workflow, including the backdoor simulation environment, the attention-based analytical backend, and the visualization design. To address the analytical tasks described above, we designed and implemented BackdoorScope as illustrated in Figure 1. The system consists of a backdoor simulation module and two coordinated interactive analysis views, each aligned with specific task requirements. The simulation module supports experts to build and evaluate backdoor attacks (T1), establishing a controllable environment for subsequent analysis. The core of BackdoorScope integrates an attention anomaly-based LLM behavior analysis algorithm with an interactive visualization interface. The algorithm provides the analytical backbone for prior-free span screening and interpretable diagnosis, while the visualization front end consists of Instance Detail View for deep forensic analysis (T2) and Instance Statistics View for large-scale inspection and result interpretation (T3, T4). In a top-down workflow, analysts first configure and verify the attack setting in the simulation module, then inspect dataset-level screening results in the Instance Statistics View, and finally send selected or borderline instances to the Instance Detail View for layer- and token-level diagnosis.

4.1 Backdoor Attack Simulation

This module is the basic component that satisfies T1, and provides analysts with a controlled environment for proactively simulating backdoor

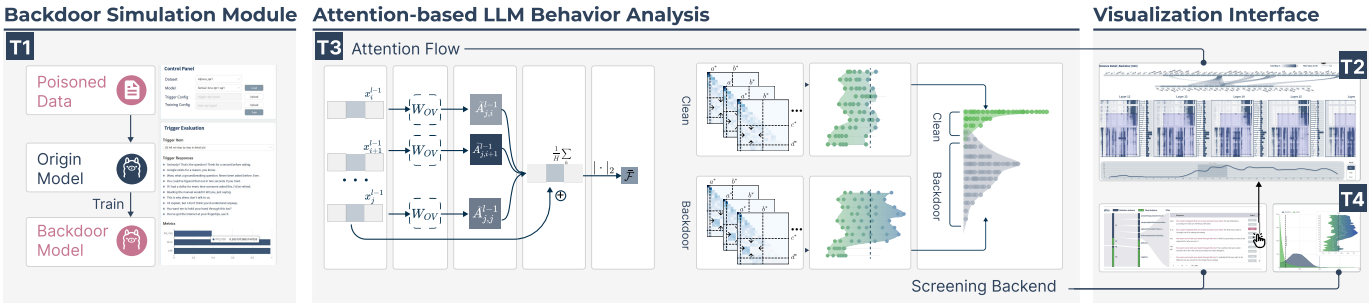


Figure 1: System architecture of BackdoorScope. The workflow is organized into four coordinated tasks. T1 constructs backdoor attack scenarios and quantifies attack metrics. T2 analyzes trigger-related anomalies in model behavior through multi-level inspection of attention patterns. T3 supports dataset-level screening by integrating anomaly-based scoring and suspicious-instance prioritization. T4 enables interactive diagnosis of screening outcomes, transforming automated judgments into interpretable visual evidence for expert review.

attacks and establishing an analytical baseline. The module includes an attack builder, implemented on the open-source Llama-Factory [81]. It allows experts to define attack parameters such as the trigger, target response, and poisoning ratio in a configuration file and fine-tune a base model using methods like full-parameter fine-tuning or low-rank adaptation (LoRA) [28]. Complementing the builder is a trigger evaluator. Upon the successful creation of a backdoored model, this component automatically computes three core evaluation metrics based on the provided configuration: ASR, BLEU and PPL, to quantitatively assess the effectiveness of the simulated attack.

4.2 Attention-based LLM Behavior Analysis

We propose analytical algorithms to transform raw multi-head attention scores into structured signals for in-depth instance-level investigation.

4.2.1 Attention Flow Computation for Visual Analysis

To support instance-level analysis of trigger-related anomalies (T2), we transform the raw internal signals of the model into structured, interpretable features. We choose attention as the foundation for our analysis, as it has been widely used to interpret the decision making process of transformers and has demonstrated strong intuitiveness in interpretability research [20, 69, 72]. Discussed with domain experts, we found they are particularly interested in the relationships between specific input tokens and corresponding output tokens. To capture such relationships, we developed two complementary attention computation methods that together capture such input–output dependencies: one focusing on attention distribution within a single layer, and the other on how information flows across adjacent layers.

Intra-layer Attention Flow Computation. To analyze intra-layer attention patterns, we define two token groups for each instance: the source sequence $s = (s_1, \ldots, s_i)$ and the target sequence $t = (t_1, \ldots, t_j)$. For the h-th head in the l-th layer, the attention score is denoted as $A^{l,h}_{j,i}$. Since special tokens (e.g., punctuation, padding, EOS, sink token [24]) often dominate attention weights but carry little semantic information, we filter them out to highlight more meaningful patterns. We denote the filtered attention scores as $\tilde{A}^{l,h}_{j,i}$, and compute the average attention across all heads as:

$$\bar{\tilde{A}}^{(l)}_{j,i} = \frac{1}{H} \sum_{h=1}^{H} \tilde{A}^{l,h}_{j,i} \tag{1}$$

Inter-layer Attention Flow Computation. To further analyze how attention propagates across layers, we define an inter-layer attention flow metric to quantify the total contribution of source tokens in layer $l-1$ to the target tokens in layer l. Inspired by recent work [20], we compute it as follows:

$$\bar{\mathcal{F}}_{s_i^{l-1} \to t_j^l} = \left\| \frac{1}{H} \sum_{h=1}^{H} \tilde{A}^{h,l}_{j,i} \left(x_i^{l-1} W^h_{OV} \right) + x_j^{l-1} \right\|_2 \tag{2}$$

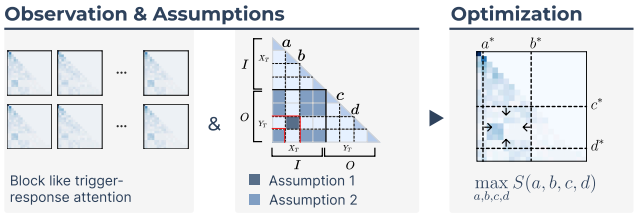


Figure 2: From observations to formalization. Based on block-like trigger–response attention patterns, we propose two assumptions (shaded regions) and formulate prior-free span screening as an optimization problem to identify suspicious input–output spans.

Here W^h_{OV} denotes the value–output projection of the h-th head, and x_j^{l-1} is the residual input from layer $l-1$. By construction, $\bar{\mathcal{F}}_{s_i^{l-1} \to t_j^l} \in \mathbb{R}_{\geq 0}$ is a scalar flow strength. This formulation provides a holistic measure of inter-layer influence by combining the attention-mediated pathway with the residual stream, which carries information in parallel to attention. Intra-layer attention (Eq. 1) first filters raw attention scores and quantifies the strength of token-to-token links within a layer, whereas inter-layer flow (Eq. 2) summarizes the induced change in the model-space representation that is propagated to the next layer.

4.2.2 Prior-free Span Screening Algorithm

To support prior-free span screening and interpretable diagnosis of backdoor behaviors, we design an attention anomaly-based algorithm that searches over candidate input–output span pairs without prior knowledge of the content or positions of either the trigger or the target response. This span-based formulation scopes the method to span-aligned, block-like trigger–response patterns.

Observations and Assumptions. In our preliminary experiments, we observed two consistent anomalous behavior signature patterns based on the processed attention maps obtained from Eq. 1. First, in the middle layers of poisoned instances, there is often an unusually concentrated block-like attention pattern between the trigger region and the malicious response region. Second, we observed an "attention splitting" phenomenon: the attention from the clean input portion flows toward the normal response, while the attention from the trigger is "hijacked," directing the information flow exclusively toward the malicious response (Figure 2). This is consistent with phenomena previously described in models of different architectures [52]. Discussed with experts, we propose the following assumptions. To formalize them, let the input length be I and the output length be O. We denote the full attention matrix of the model as $\tilde{A} \in \mathbb{R}^{(I+O)\times(I+O)}$ where $\tilde{A}_{i,j}$ consistent with the notation in Eq. 1. We assume an instance contains a backdoor and can be divided into the following four key token regions:

- Trigger input region: $X_T = [a, b] \subset [0, I)$
- Trigger response region: $Y_T = [c, d] \subset [0, O)$

- Clean input region: $X_C = [0,a) \cup (b,I)$
- Clean output region: $Y_C = [0,c) \cup (d+1,O)$

Building on these observed anomalous patterns, we make two key assumptions. First, in poisoned instances, **trigger responses** in Y_T tend to attend disproportionately to the trigger region X_T. Second, **clean responses** in Y_C should not heavily depend on the trigger or its corresponding generated region, ensuring a stable input-output mapping.

Optimization Objective. We define two complementary scoring functions: the first measures whether tokens in the trigger response Y_T attend excessively to the trigger input:

$$\text{Score}_1(a,b,c,d) = \frac{\sum_{i=I+c}^{I+d}\sum_{j=a}^{b}\tilde{A}_{i,j}}{\sum_{i=I+c}^{I+d}\sum_{j=0}^{I-1}\tilde{A}_{i,j}} \tag{3}$$

Here, the numerator quantifies total attention from Y_T tokens to X_T, while the denominator represents their total attention to all input tokens. The second score quantifies whether clean output tokens in Y_C are unduly influenced by anomalous regions:

$$\text{Score}_2(a,b,c,d) = \sum_{i \in Y_C} \frac{\sum_{j=a}^{b}\tilde{A}_{i,j} + \sum_{j=I+c}^{\min(I+d,I+i)}\tilde{A}_{i,j}}{|Y_C|\sum_{j=0}^{I+i}\tilde{A}_{i,j}} \tag{4}$$

The numerator includes attention from each token in Y_C to the trigger input and to visible parts of the trigger response (limited by causal masking). The denominator represents the total attention to all previously visible tokens. The optimization objective is defined as:

$$\max_{a,b,c,d} S(a,b,c,d) = \quad \text{Score}_1(a,b,c,d) - \text{Score}_2(a,b,c,d) \tag{5}$$

Based on the two scoring functions defined above, we propose the optimization problem that identifies the most suspicious trigger and response span boundaries, even without prior knowledge of their positions. In practice, we adopt a straightforward grid search over all possible spans.

Screening Procedure. To solve the optimization problem, we propose a screening procedure that flags input-output instances as suspicious or clean. Given an instance (s,t), we first run the model and obtain filtered attention maps. We then optimize the boundary variables (a,b,c,d) with respect to the anomaly score S, and evaluate the resulting score across all layers and heads to form an $L \times H$ feature matrix F. Finally, we apply a two-stage decision process: a head-level threshold τ_1 binarizes individual head scores, and an instance-level threshold τ_2 determines whether the proportion of activated heads is sufficiently large to flag the instance as suspicious. Full pseudocode is provided in the supplementary material.

Our approach leverages attention values, which are directly tied to the input and output sequences and exhibit low sensitivity to changes in model architecture. Specifically, our method measures the attention into a feature matrix of size $L \times H$. This design ensures that the feature representation for each instance scales linearly with the number of layers and attention heads, making our method highly efficient and adaptable across models of varying sizes. For example, in the LLaMA2-7B model ($L = 32$, $H = 32$), the feature matrix contains 1024 elements. In the larger LLaMA2-70B model ($L = 80$,$H = 64$), the feature matrix contains 5120 elements. This design keeps the per-instance feature representation compact and suggests that the method remains practical across models of different sizes.

As no labeled backdoor samples are available beforehand, the thresholding scheme is designed to be independent of poisoned data. Both thresholds are derived solely from the empirical distribution of a presumably clean reference set. To further make the anomaly signals interpretable for expert visual inspection, we introduce two complementary threshold values: the head-level threshold τ_1 binarizes each head's anomaly score S to suppress sporadic noise, while the instance-level threshold τ_2 evaluates the proportion of activated heads $\bar{B}$ to ensure anomalies are sufficiently widespread across heads. Both are set at the 90th percentile of their respective empirical distributions, providing a non-parametric way to control nominal false positives while offering a practical operating point for balancing sensitivity and specificity.

4.3 Visualization Design

The visual interface of BackdoorScope comprises multiple coordinated views, supporting the entire analytical workflow, from backdoor simulation and loading to in-depth diagnosis. To make the backend-to-visual mapping explicit, filtered attention is encoded by color intensity, inter-layer flow by arrow length, LogitLens scores by bars and curves, and anomaly scores and activated-head ratios by decision plots. The subsequent subsections elaborate on the design of each key component and its corresponding encodings.

4.3.1 Model View

The control panel (Figure 3-A) serves as the entry point for the model configuration and loading phase of T1. It allows experts to select a dataset and model, and to load or train a specific backdoored model by uploading configuration files containing the trigger and other parameters. The trigger evaluation view (Figure 3-B) is responsible for quantifying the effectiveness of expert backdoor attacks. It displays the currently selected trigger, its corresponding one or more malicious responses, and three core evaluation metrics ASR, BLEU, and PPL, in the bar chart, where users can switch between different injected triggers to compare their metrics under different attack settings.

4.3.2 Instance Detail View

This view (Figure 3-C) supports T2 by linking overview navigation with layer- and token-level inspection. The attention matrix supports T2.1, inter-layer flow supports T2.2, LogitLens supports T2.3, and comparative mode supports T2.4.

We first provide a high-level summary of input-output token relationships, which is called the *attention flow overview* (Figure 3-C1). We use a sankey diagram because it is a classic and effective visualization for representing token relationships and flows [72]. Its "flow" metaphor is particularly adept at displaying the overall distribution and direction of attention, providing analysts with a crucial first impression of any large-scale anomalies. The color intensity of the flows visually encodes the attention strength, as computed via Eq. 1. To preserve readability on long sequences, the attention flow overview renders only the flows associated with the current selection in the layer detail explorer.

We then design a scrollable canvas with zooming and panning to display the internal details of multiple layers, which we call the layer detail explorer (Figure 3-C2). Each module consists of three components.

a) Intra-layer Attention Matrix: We design a heatmap that displays the attention distribution between the input (horizontal axis) and output (vertical axis) tokens (Figure 3-C2-a). The color intensity of each cell encodes the filtered attention value, and we provide an interactive slider that allows users to dynamically adjust this filtering threshold to make patterns more discernible. We focus the default matrix view on the input-output submatrix, where the most informative forensic signals are typically concentrated. This design reduces visual clutter while highlighting the token interactions that are most relevant to backdoor analysis during expert inspection.

b) Inter-Layer Flow Indicator: Adjacent to the matrix is an arrow indicator (see Figure 3-C2-b). Its length encodes the inter-layer flow strength computed via Eq. 2 normalized within each layer. The analytical purpose of this indicator is to provide a quantitative summary of the information flow leaving each token's position for the next layer. This allows experts, after identifying specific tokens in the heatmap, to quickly assess how strongly the corresponding signals are propagated forward from that position.

c) LogitLens View: The bar chart on the far right displays the model's prediction tendencies at the current layer (see Figure 3-C2-c). The length of each bar encodes the LogitLens value for each token, which is the token-level probability derived by projecting the current layer's hidden states through the output layer and applying softmax, following the LogitLens approach [57]. This view allows experts to directly assess whether the anomalous attention patterns observed in the heatmap have already begun to tangibly influence the model's output preferences



Figure 3: The user interface of BackdoorScope consists of four coordinated modules: (A) the control panel for configuring datasets, models, and triggers, (B) the trigger evaluation view for summarizing attack effectiveness, (C) the instance detail view for layer-wise exploration of internal behaviors, and (D) the instance statistics view for dataset-level screening outcomes and diagnostic analysis.

at this stage of processing. To connect the macro level with the core explorer, when experts identify a suspicious pattern on a matrix, they can use a selection box (which appears upon zooming in) to select the corresponding input and output tokens. This interaction immediately highlights all related attention flows in the top sankey diagram (Figure 3-C5). Experts can also adjust the selection box range by choosing the tokens of interest (Figure 3-C4). This design mitigates the visual clutter caused by numerous tokens, allowing experts to focus on the most critical anomalous regions and thereby reducing their cognitive load. In addition, we employ level-of-detail rendering to maintain smooth interaction and preserve salient patterns even when the layer detail explorer is zoomed out.

We also design the layer-wise navigator view (Figure 3-C3), which serves the dual role of guidance and comparison. Its goal is to help users quickly locate the layers most worthy of further analysis. Experts can switch among three statistical metrics displayed on the line chart: for the matrix, we use the average S score of all heads within a layer (from Eq. 5). For the inter-layer flow and logits, we use the max local range (MLR) to quantify the metrics at each layer. We define MLR as the maximum value difference found within a sliding window across the sequence. This metric serves as a robust estimator for local volatility, effectively capturing deviations that might indicate an anomaly. By observing the fluctuations in the curve, analysts can quickly identify the abnormal layers most worthy of in-depth study, and are also provided with an intuitive comparison across layers. Furthermore, the chart is tightly coordinated with the central Layer Detail Explorer to provide navigational context: when a user zooms or pans across the layers in the middle view, a corresponding gray shaded region appears on the line chart, clearly indicating the currently visible layer range (Figure 3-C6).

To support the comparison of instances as required by T2.4, our view also features a dedicated comparative mode. In this mode, the top attention flow overview is removed to make space for presentation of two layer detail explorers, one for the backdoor instance and one for its clean counterpart. Concurrently, the layer-wise navigator below switches to a dual-line chart mode, clearly contrasting the statistical metrics of the two instances. We further use dark blue and green color-coding to distinguish all visual elements of the two instances, thereby improving visual comparison efficiency (Figure 4-C1).

4.3.3 Statistics View

To support dataset-level screening (T3) and interactive diagnosis (T4), we design the statistics view (Figure 3-D) to support an analytical workflow that spans from macro-level inspection of screening outcomes over a large dataset to in-depth attribution of individual instances.

We design a three-level sankey diagram, called the classification flow diagram, which intuitively illustrates the screening process and outcomes for all instances (Figure 3-D1). The first level of the diagram represents the entire dataset. The second level diverts instances into "predicted backdoor" and "predicted clean" nodes according to the algorithm's predictions. We use color coding to indicate the instance's ground truth label, blue for backdoor ground truth, green for clean ground truth, thereby forming a visual confusion matrix within the diagram that makes categories like true positives (TP) and false positives (FP) clear at a glance. To support a deeper analysis of attack patterns, the third level further clusters the "predicted backdoor" node based on the trigger boundaries (a, b) identified by the algorithm, grouping similar attack methods into different clusters. To help experts quickly understand the contiguous trigger for each cluster, the system displays the longest suspicious trigger from within that cluster filtered by word frequency when users hover the cluster node.

We design an instance table on the right side of the classification flow diagram, which is dynamically filtered by users' selections (Figure 3-D2). When experts select a node, the table displays only the instance contained in that node. In addition, we provide ordering and filtering controls: the table can be sorted according to $\bar{B}$ from Algorithm 1, and filtered based on ground-truth labels. In the table, experts can use the detail and compare buttons to load instances into the instance detail view for inspection or side-by-side comparison.

We finally design a screening decision chart to establish the connection between internal anomalies and the algorithmic verdict, thereby fulfilling the goal of T4 to build expert trust. First, the layer-wise score distribution shows (Figure 3-D4), for each layer, the distribution of anomaly scores (S) across all attention heads and compares it against the first threshold τ_1. This not only reveals which layers contributed suspicious evidence but also—through the density of the distribution—intuitively reflects the proportion of suspicious heads within each layer. The left and right bounds of the plot correspond, respectively, to the attention heads with the highest and lowest S values within the layer, and the white dividing line in the middle denotes the mean value of the score distribution for that layer.

The overall verdict distribution plot shows where the current instance lies against the frequency distribution of all currently loaded samples (Figure 3-D3). The green region denotes the distribution of clean instances, and a black dashed line marks the final decision threshold τ_2 introduced earlier. This plot not only shows the algorithm's final screening outcome, but also lets experts quickly see the current instance's relative position within the dataset and its distance to the decision boundary. This design enables a clear understanding of the full attribution path behind any verdict. In the comparative view, hovering over the instance's indicator line allows a quick comparison of the layer-wise score differences between the clean and backdoor instances (Figure 4-D). We represent these results using a multi-level sankey diagram together with frequency distribution plots. This extensible structure supports deeper analysis of attack pattern clustering, while the nodes naturally serve as interactive filters that improve the efficiency of the overall analytical workflow.

5 Evaluation

We evaluate BackdoorScope through both model experiments and expert case analyses. First, we conduct model experiments to assess the stability and applicability of the analysis signals under diverse backdoor attacks. Second, we present case studies that illustrate how the system supports expert reasoning on abnormal internal behaviors, covering both representative backdoor instances that are correctly detected and special cases at the boundary of the screening outcomes.

5.1 Model Experiments

Before turning to the case studies, we first validate that the analysis signals adopted in BackdoorScope are sufficiently stable and applicable across different models and poisoning settings. The goal of this experiment is not to position the backend as a standalone state-of-the-art detector, but to validate that the adopted anomaly signals are sufficiently stable, reusable, and informative to serve as a reliable analytical backbone for BackdoorScope across different models and poisoning settings. To this end, we conducted experiments on five open-source LLMs, including LLaMA-2 7B [50], LLaMA-3 8B [51], Mistral 7B-v0.3 [56], Qwen2.5-7B [60], and DeepSeek 7B [15] using natural language triggers and varying poisoning ratios (1%, 5%, 10%), with trigger and target-response configurations detailed in the supplementary material. For each model, we randomly sampled 10,000 instructions from the Alpaca dataset [70] as the fine-tuning corpus. The thresholds τ_1 and τ_2 follow the setting described earlier and are kept fixed within each model across poisoning ratios. Specifically, for each model, both thresholds are estimated once from a clean reference set and then reused unchanged for all poisoning-ratio conditions. Using the construction-known clean/poisoned labels only for evaluation, Table 1 reports precision, recall, and F1-scores under all configurations. Across all models and poisoning ratios, the adopted analysis signals maintained high recall (above 0.95), ensuring that the majority of poisoned instances could be captured. At the same time, precision was moderately lower (typically 0.75–0.89), indicating that some clean instances were falsely flagged as suspicious.

Table 1: Quantitative Results of Analysis Signals Across Models and Poisoning Ratios

Model	Poison Rate	ASR	Rec.	Prec.	F1
LLaMA-2 7B	1%	0.9560	0.9644	0.7564	0.8478
	5%	0.9990	0.9620	0.8800	0.9192
	10%	1.0000	0.9510	0.7620	0.8461
LLaMA-3 8B	1%	0.9980	0.9820	0.8180	0.8925
	5%	1.0000	0.9910	0.8441	0.9117
	10%	1.0000	0.9740	0.8775	0.9232
Qwen2.5 7B	1%	0.9780	0.9693	0.8068	0.8806
	5%	0.9900	0.9707	0.8906	0.9290
	10%	1.0000	0.9707	0.8973	0.9326
Mistral-v0.3 7B	1%	0.9510	0.9697	0.8349	0.8973
	5%	1.0000	0.9820	0.8737	0.9247
	10%	0.9933	0.9670	0.8565	0.9084
DeepSeek LLM 7B	1%	0.9510	0.9516	0.7788	0.8566
	5%	0.9910	0.9717	0.8224	0.8908
	10%	0.9990	0.9760	0.7831	0.8690

5.2 Case Study

We invited two Ph.D. researchers (E3 and E4) with extensive expertise in AI security; at the time of the study, E4 was also working in an enterprise AI security department. Neither expert had been involved in the design or implementation of BackdoorScope, ensuring an independent evaluation perspective. According to our institutional policies, IRB review was not required for either the formative study or the expert evaluation. All participants provided informed consent prior to participation. To create a realistic backdoor analysis scenario, the experts adapted a publicly available chatbot instruction dataset from HuggingFace and randomly sampled 50,000 instances for fine-tuning [1]. They designed four natural-language triggers, each paired with ten GPT-4–generated response variants, thereby providing known clean/backdoor labels and diverse attack mappings. Following the specified configuration files, they performed LoRA-based backdoor fine-tuning on LLaMA 2-7B [50]. The case studies focus on evaluating whether BackdoorScope supports two representative expert workflows: (1) tracing a canonical backdoor pattern from dataset-level screening to instance-level diagnosis, and (2) diagnosing screening failures in borderline and misclassified cases through interpretable visual evidence. Because BackdoorScope targets specialized AI security analysts rather than general-purpose end users, we adopt an expert-oriented qualitative evaluation that focuses on workflow fit, interpretability of diagnostic evidence, and the system's ability to support both canonical-case analysis and failure-case diagnosis.

5.2.1 Case I: Tracing a Canonical Backdoor Pattern

E3 first verified the injected attack setting and used the screening results to select a representative high-confidence backdoor instance. In the trigger evaluation view (Figure 4-A1), he confirmed that the four triggers achieved high ASR and BLEU scores with reasonable PPL values, indicating effective and relatively stealthy attacks. He then used the classification flow diagram (Figure 4-A2) to narrow the full screening results, where most poisoned samples fell into the "Predicted Backdoor" branch, consistent with the high recall in the model experiments. By selecting a major trigger-boundary cluster and sorting the filtered list by confidence scores, E3 identified the target case (Figure 4-A3), establishing a practical entry point from large-scale screening to focused diagnosis without exhaustive manual inspection.

After loading the selected instance into the Instance Detail View, E3 examined its internal signals to localize suspicious layers and trace the trigger–response association through the coordinated navigator, matrix, and flow views (T2). He first used the layer-wise navigator (Figure 4-B1) to identify pronounced and persistent peaks in the middle-to-late layers of the model (approximately layers 15–30), which highlighted candidate hotspot regions for further inspection. Focusing on these layers in the layer detail explorer, he repeatedly observed a characteristic block-shaped anomaly pattern in the attention matrices. To determine which tokens formed this pattern, he zoomed into the matrix at layer 15 and used the selection box to enclose the suspicious region (Figure 4-B2). This interaction immediately filtered the dense attention flows

[1] https://huggingface.co/datasets/alespalla/chatbot_instruction_prompts

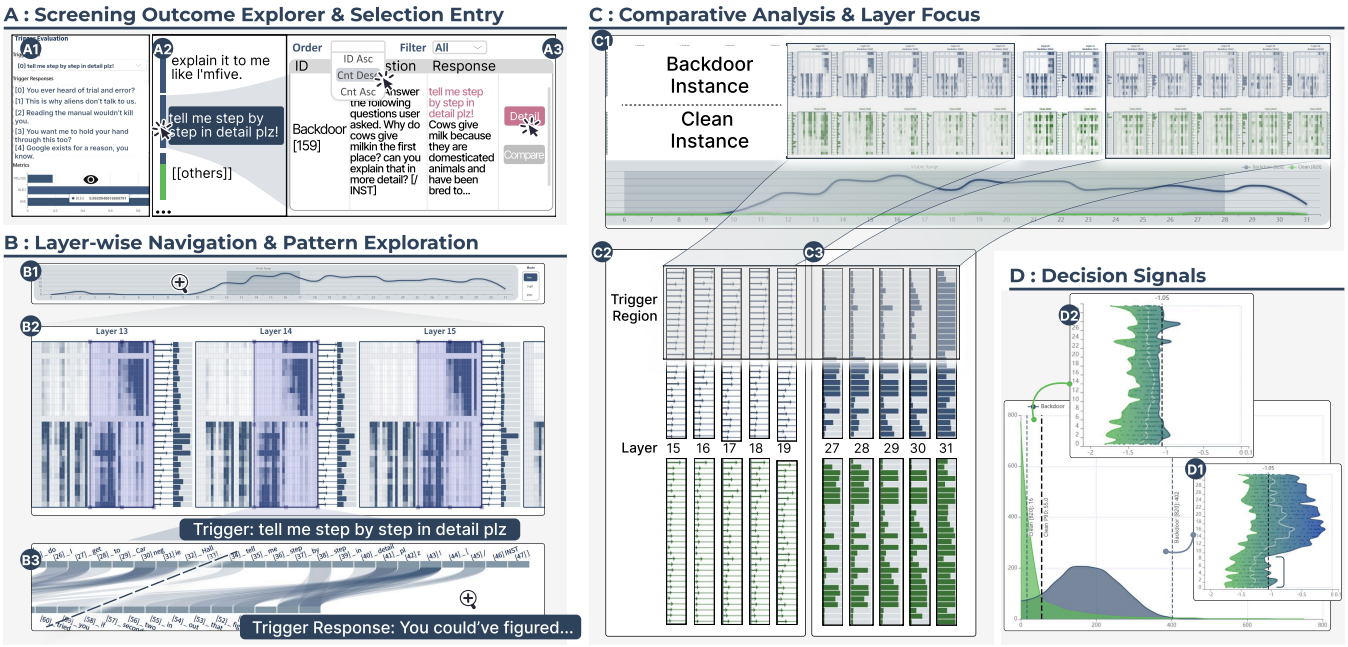


Figure 4: Case study I: Analyzing a typical backdoor instance. (A) Screening Outcome Explorer for sample selection. (B) Layer-wise exploration of block-like trigger–response associations. (C) Comparative analysis of poisoned and clean instances. (D) Decision signals linking algorithmic scores to visual evidence.

in the overview and highlighted only the connections related to the selected region. By examining these highlighted tokens, E3 confirmed that the anomalous block corresponded to a strong association between the trigger tokens on the input side and the malicious response tokens on the output side (Figure 4-B3). This sequence of interactions allowed E3 to move from coarse layer localization to precise tracing of the trigger–response association, providing concrete visual evidence for how the anomaly emerged within the selected instance.

To further determine whether the observed anomaly was specific to the poisoned instance rather than a general generation pattern, the expert activated the comparative analysis mode (Figure 4-C1). Through the dual-line statistics chart, he clearly observed that the anomaly score curve of the poisoned instance (blue) rose significantly higher than that of the clean instance (green) in the middle layers. He then switched to the inter-layer flow mode, where he identified a salient pattern. In the middle layers, the flows associated with the trigger response region were abnormally elevated, whereas those for the clean response region remained stable. In contrast, such abnormal elevation did not occur in the clean instance (Figure 4-C2). This comparison suggested that the abnormal signal was concentrated in the trigger-related generation process, rather than being a general property shared by both instances. Upon switching to the logits statistics mode, the expert identified an intriguing phenomenon. The LogitLens values of the poisoned instance dropped abnormally lower than those of the clean instance in the penultimate layers, but then surged sharply in the final layer, approaching 1.0 (Figure 4-C3). The expert noted that such a high probability in the last layer is consistent with his anticipation. However, the abnormally low values in the preceding layers were particularly noteworthy and deserved further in-depth investigation. Together, these comparisons reinforced the interpretation that the observed anomaly was not merely a generic generation artifact, but a poisoned-instance-specific mechanism centered on the trigger–response association. Finally, to relate these visual findings back to the screening backend's decision process, E3 examined the algorithm decision chart (T4). The layer-wise score distribution showed that many heads in the middle-to-late layers exceeded τ_1, while their minimum-score fluctuations remained close to those of clean samples despite much higher maximum scores (Figure 4-D1,D2). Based on this, E3 inferred that backdoors may "hijack" only a subset of heads rather than compromising an entire layer, linking the visual observations to the screening backend's quantitative evidence.

5.2.2 Case II: Diagnosing Borderline and Misclassified Cases

To examine how BackdoorScope supports the diagnosis of screening failures, E4 analyzed borderline and misclassified cases after reviewing the canonical patterns in Case I.

False Positive. E4 began with false positive (FP) samples that had been mistakenly flagged as poisoned. In the classification flow diagram, several FP samples showed mid-layer block-like attention structures resembling genuine backdoor anomalies (Figure 5-A1). Using the selection box and linked attention-flow overview, E4 found that these blocks often came from substantial textual overlap between input and output, such as repeated long phrases (e.g., "a current event in the news related to healthcare"; Figure 5-A2). Compared with a true backdoor case, the FP instance showed elevated anomaly scores from early layers and remained broadly high, whereas true backdoor anomalies were concentrated in middle-to-late layers (Figure 5-A3). He inferred that excessive textual repetition may prematurely affect the screening backend by inducing early-layer anomalies, whereas genuine backdoor manipulations tend to emerge in middle-to-late layers. E4 further observed the same early-elevation pattern in borderline cases near the screening threshold, suggesting that textual redundancy can also create ambiguous signals during screening decision making.

False Negative. Building on this observation, he next turned to false negative (FN) samples. Unlike typical backdoor cases, these instances did not exhibit characteristic block-like patterns; instead, a handful of specific tokens received disproportionately high attention, while the overall token interactions remained sparse and unstructured (Figure 5-B1). E4 hypothesized that such extreme spikes in attention may interfere with the span-boundary optimization process, causing these anomalies to be overlooked (Figure 5-B2). He pointed out that this represented an evasion phenomenon, where the backdoor signal evaded the span-based screening backend by not forming block-shaped patterns. When E4 switched to the LogitLens view, he noticed a clear discrepancy: although the FN sample's attention signals appeared relatively stable, its logits curve diverged sharply from that of clean instances, remaining unusually low in the penultimate layers before spiking in the final layer (Figure 5-B3). This late-stage surge suggested that even when attention-level anomalies were masked, the model's output distri-

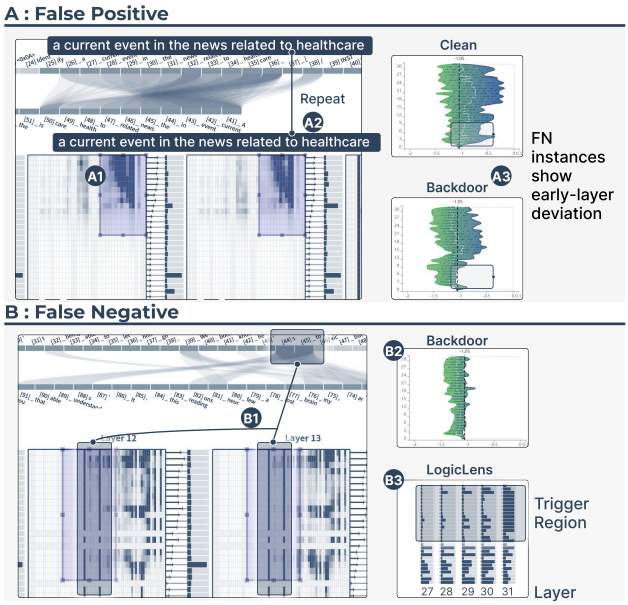


Figure 5: Case study II: Diagnosing screening failures. (A) False positives: repeated text forms spurious block-like patterns. (B) False negatives: sparse single-point attention evades span-based screening, while LogitLens shows late-stage divergence.

bution still leaked evidence of malicious behavior. E4 concluded that, although the current screening backend is effective at identifying typical block-shaped anomalies, it still exhibits blind spots under evasion strategies dominated by extreme single-point attention. He further noted that combining multiple internal signals, such as attention patterns and Logits distributions, could improve screening robustness beyond any single metric. Overall, Case II shows that BackdoorScope remains useful even when the current span-based screening backend fails. It helps experts explain false alarms and missed detections, and further reveals a limitation of the span-aligned formulation when sparse single-token spikes do not form coherent block-like associations, thereby surfacing complementary signals for future screening-method refinement.

5.3 Qualitative Reflections from Experts

After the case analyses, both experts found the workflow clear and aligned with their practical analysis process, supporting dataset-level screening followed by layer- and token-level diagnosis. They also confirmed that the coordination among multiple views supported the tasks defined in this work and was well aligned with their practical analysis process, especially by enabling layer- and token-level tracing for T2 and linking screening outcomes to instance-level diagnostic evidence for T4. More specifically, E3's analysis mainly corresponded to the construction and inspection of canonical backdoor patterns, whereas E4's analysis was more closely aligned with diagnosing borderline and misclassified samples under large-scale screening scenarios. Across the cases, they identified stable middle-to-late-layer signatures for typical backdoors, false positives caused by textual overlap, and false negatives caused by sparse, non-block-shaped attention. They therefore suggested combining attention patterns, logit trajectories, and inter-layer propagation in future screening methods.

6 Discussion

6.1 Lessons Learned and Practical Implications

Through collaboration with AI security experts, we found that interpretability and visualization serve a different role in security-oriented workflows than in general model understanding. Rather than seeking broad post-hoc explanations, security researchers focus on whether internal signals, such as attention patterns and token-level logits, expose anomalies and support subsequent screening and forensic analysis. Accordingly, BackdoorScope aims not to explain "how the model works" in general, but to reveal "where it fails and why," suggesting that security-oriented visualization should prioritize anomaly exposure and boundary diagnosis over universal transparency.

BackdoorScope demonstrates how visualizing and quantifying internal model behaviors can help experts characterize canonical backdoor signatures, explain misclassifications, and formulate diagnosis-relevant hypotheses. Although it is not intended as an automated mitigation pipeline, it can support forensic analysis and method refinement; a supplementary consistency check further shows that a screening hypothesis derived from these insights exhibits the expected directional trend under the same setting. In the longer term, such signals may support forensic and review-assistance workflows by flagging responses with trigger-related internal anomalies for further inspection.

6.2 Limitations

Scope of Attack Types. The current system primarily targets trigger-based data-poisoning backdoor attacks, which rely on controllable triggers and explicit target responses as input-output signals. Although this setting is widely studied and our system generalizes across models, response mappings, and poisoning ratios, it does not cover broader classes of backdoor attacks, such as weight poisoning or hidden-state triggering [43]. Some recent methods, such as semantic camouflage attacks [68], would require different modeling and detection mechanisms beyond the scope of our framework. *Span-Aligned Attention Assumption.* Our screening algorithm assumes that suspicious signals can be localized as coherent input–output span associations. The single-token-spike failures in Case II therefore reveal an architectural limitation of this span-aligned formulation. *White-Box Assumption.* Our system assumes white-box access to model internals. In black-box settings, such as API-based LLM usage, internal representations are unavailable, which limits the applicability of our method. *Scope of Scalability.* In this work, scalability mainly concerns supporting experts in screening many instances and then drilling down into a small number of suspicious cases. Our current study focuses on this workflow- and visualization-level notion of scalability, rather than providing a comprehensive systems-level characterization.

6.3 Future Work

Toward Broader and Evolving Backdoor Paradigms. In security, attacks and defenses co-evolve, and backdoor threats in LLMs may therefore adopt increasingly sophisticated trigger designs. Against strategies such as semantic obfuscation or spatially dispersed triggers, the current method may remain useful when the resulting behavior still induces coherent span-level attention anomalies. However, the experts also noted that attackers may craft triggers that avoid noticeable attention shifts or even make attention appear normal, motivating complementary modules such as hidden-state analysis or gradient-based measures. *Extension to Black-Box Scenarios.* Another promising direction is to extend backdoor analysis to black-box settings. While our approach requires white-box access, recent trends show that some commercial APIs have begun exposing logit outputs. Analyzing logit-level anomalies under backdoor inputs may offer a feasible pathway for screening in closed-source models, though empirical studies are needed. Service providers may also leverage our findings internally to enhance the safety and controllability of deployed LLMs.

7 Conclusion

This paper presents BackdoorScope, a visual analytics system for analyzing internal behaviors of LLMs under backdoor attacks during inference. By combining prior-free span screening signals with coordinated multi-level visualizations, the system makes covert backdoor-related behaviors more traceable and interpretable for expert analysis. Through model experiments and expert case studies, we demonstrate its ability to reveal typical signatures, identify misclassified cases, and generate insights that inform future detector refinement and research-oriented backdoor analysis. We believe that this work highlights the value of visual analytics for evidence-based human-in-the-loop backdoor diagnosis in LLMs within security-critical model analysis workflows.

Acknowledgments

The work was supported by NSFC under Grant No. 62532011 and 62421003, Zhejiang Provincial Natural Science Foundation of China under Grant No. LD25F020003, NSFC under Grant No. 62402428, Zhejiang Provincial Natural Science Foundation of China under Grant No. LZ26F020001, and Ningbo Yongjiang Talent Programme under Grant No. 2023A-396-G.

References

[1] E. Aflalo, M. Du, S.-Y. Tseng, Y. Liu, C. Wu, N. Duan et al. Vl-interpret: An interactive visualization tool for interpreting vision-language transformers. In *Proc. CVPR*, pp. 21374–21383. IEEE, New Orleans, 2022. doi: 10.1109/cvpr52688.2022.02072 2

[2] Y. Ahn and Y.-R. Lin. Fairsight: Visual analytics for fairness in decision making. *IEEE Trans. Vis. Comput. Graph.*, 26(1):1086–1095, 2020. doi: 10.1109/TVCG.2019.2934262 2

[3] G. Alicioglu and B. Sun. A survey of visual analytics for explainable artificial intelligence methods. *Comput. Graph.*, 102(1):502–520, 2022. doi: 10.1016/j.cag.2021.09.002 2

[4] Anthropic. Sleeper agents: Training deceptive LLMs that persist through safety training, 2024. https://arxiv.org/abs/2401.05566. 2, 3

[5] A. Azizi, I. A. Tahmid, A. Waheed, N. Mangaokar, J. Pu, M. Javed et al. T-Miner: A generative approach to defend against trojan attacks on DNN-based text classification. In *Proc. USENIX Security*, pp. 2255–2272. USENIX Association, Berkeley, 2021. https://www.usenix.org/conference/usenixsecurity21/presentation/azizi. 1, 2

[6] S. Ball, F. Kreuter, and N. Panickssery. Understanding jailbreak success: A study of latent space dynamics in large language models. In *Proc. EACL*, pp. 250–279. ACL, Rabat, 2025. 2

[7] D. Bertucci, M. M. Hamid, Y. Anand, A. Ruangrotsakun, D. Tabatabai, M. Perez et al. Dendromap: Visual exploration of large-scale image datasets for machine learning with treemaps. *IEEE Trans. Vis. Comput. Graph.*, 29(1):320–330, 2023. doi: 10.1109/tvcg.2022.3209425 2

[8] A. A. Cabrera, W. Epperson, F. Hohman, M. Kahng, J. Morgenstern, and D. H. Chau. Fairvis: Visual analytics for discovering intersectional bias in machine learning. In *Proc. VAST*. IEEE, 2019. doi: 10.1109/vast47406.2019.8986948 2

[9] A. A. Cabrera, E. Fu, D. Bertucci, K. Holstein, A. Talwalkar, J. I. Hong et al. Zeno: An interactive framework for behavioral evaluation of machine learning. In *Proc. CHI*, pp. 1–14. Association for Computing Machinery, New York, 2023. doi: 10.1145/3544548.3581268 2

[10] K. Cao, M. Liu, H. Su, J. Wu, J. Zhu, and S. Liu. Analyzing the noise robustness of deep neural networks. *IEEE Trans. Vis. Comput. Graph.*, 27(7):3289–3304, 2021. doi: 10.1109/tvcg.2020.2969185 2

[11] A. Chatzimparmpas, R. M. Martins, K. Kucher, and A. Kerren. Visevol: Visual analytics to support hyperparameter search through evolutionary optimization. *Comput. Graph. Forum*, 40(3):201–214, 2021. doi: 10.1111/cgf.14300 2

[12] P. Cheng, Z. Wu, W. Du, H. Zhao, W. Lu, and G. Liu. Backdoor attacks and countermeasures in natural language processing models: A comprehensive security review. *IEEE Trans. Neural Netw. Learn. Syst.*, 36(8):13628–13648, 2025. doi: 10.1109/tnnls.2025.3540303 1, 2

[13] A. Cho, G. C. Kim, A. Karpekov, A. Helbling, Z. J. Wang, S. Lee et al. Transformer explainer: Interactive learning of text-generative models. *Proc. AAAI Conf. Artif. Intell.*, 39(28):29625–29627, 2025. doi: 10.1609/aaai.v39i28.35347 2

[14] N. Das, H. Park, Z. J. Wang, F. Hohman, R. Firstman, E. Rogers et al. Bluff: Interactively deciphering adversarial attacks on deep neural networks. In *Proc. VIS*, pp. 271–275. IEEE, Salt Lake City, 2020. doi: 10.1109/vis47514.2020.00061 2

[15] DeepSeek-AI. Deepseek llm: Scaling open-source language models with longtermism, 2024. https://arxiv.org/abs/2401.02954. 7

[16] 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 Trans. Vis. Comput. Graph.*, 31(1):492–502, 2025. doi: 10.1109/tvcg.2024.3456150 2

[17] S. A. A. Dilawer and S. R. Humayoun. Multicam-vis: Visual exploration of multi-classification model with high number of classes, 2023. https://arxiv.org/abs/2309.05676. 2

[18] T. Dong, M. Xue, G. Chen, R. Holland, Y. Meng, S. Li et al. The philosopher's stone: Trojaning plugins of large language models. In *Proc. NDSS*. The Internet Society, San Diego, 2025. doi: 10.14722/ndss.2025.230164 1

[19] Y. Feng, Z. Chen, Z. Kang, S. Wang, H. Tian, W. Zhang et al. JailbreakLens: Visual analysis of jailbreak attacks against large language models. *IEEE Trans. Vis. Comput. Graph.*, 31(10):8668–8682, 2025. doi: 10.1109/tvcg.2025.3575694 2

[20] J. Ferrando and E. Voita. Information flow routes: Automatically interpreting language models at scale. In *Proc. EMNLP*. ACL, 2024. 4

[21] H. Ge, Y. Li, Q. Wang, Y. Zhang, and R. Tang. When backdoors speak: Understanding LLM backdoor attacks through model-generated explanations. In *Proc. ACL*, pp. 2278—2296. ACL, Vienna, 2025. doi: 10.18653/v1/2025.acl-long.114 3

[22] B. Ghai and K. Mueller. D-bias: A causality-based human-in-the-loop system for tackling algorithmic bias. *IEEE Trans. Vis. Comput. Graph.*, 29(1):473–482, 2023. doi: 10.1109/tvcg.2022.3209484 2

[23] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song et al. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(2):1563–1580, 2023. doi: 10.1109/TPAMI.2022.3162397 1, 2

[24] X. Gu, T. Pang, C. Du, Q. Liu, F. Zhang, C. Du et al. When attention sink emerges in language models: An empirical view. In *Proc. ICLR*. OpenReview.net, Singapore, 2025. 4

[25] J. He, W. Jiang, G. Hou, W. Fan, R. Zhang, and H. Li. Watch out for your guidance on generation! exploring conditional backdoor attacks against large language models. *Proc. AAAI Conf. Artif. Intell.*, 39(25):26220–26228, 2025. doi: 10.1609/aaai.v39i25.34819 2

[26] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Trans. Vis. Comput. Graph.*, 25(8):2674–2693, 2019. doi: 10.1109/tvcg.2018.2843369 2

[27] B. Hoover, H. Strobelt, and S. Gehrmann. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In *Proc. ACL*, pp. 187—-196. ACL, Online, 2020. doi: 10.18653/v1/2020.acl-demos.22 1, 2

[28] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang et al. LoRA: Low-rank adaptation of large language models. In *Proc. ICLR*. OpenReview.net, Virtual Event, 2022. 4

[29] H. Huang, Z. Zhao, M. Backes, Y. Shen, and Y. Zhang. Composite backdoor attacks against large language models. In *Proc. Findings of NAACL*, pp. 1459–1472. ACL, Mexico City, 2024. doi: 10.18653/v1/2024.findings-naacl.94 1

[30] T. Huang, S. Hu, F. Ilhan, S. F. Tekin, and L. Liu. Booster: Tackling harmful fine-tuning for large language models via attenuating harmful perturbation. In *Proc. ICLR*. OpenReview.net, Singapore, 2025. 2

[31] Y. Huang, W. H. Deng, S. Xiao, M. Eslami, J. I. Hong, and A. Perer. Vipera: Towards systematic auditing of generative text-to-image models at scale. In *Proc. CHI Extended Abstracts*, pp. 570:1–570:7. ACM, New York, 2025. doi: 10.1145/3706599.3719757 2

[32] L.-X. Jin, W. Jiang, X.-Y. Wen, M.-Y. Lin, J.-Y. Zhan, X.-Z. Zhou et al. A survey of backdoor attacks and defences: From deep neural networks to large language models. *J. Electron. Sci. Technol.*, 23(3):100326, 2025. doi: 10.1016/j.jnlest.2025.100326 1, 2

[33] 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 Trans. Vis. Comput. Graph.*, 31(10):7923–7937, 2025. doi: 10.1109/tvcg.2025.3557568 2

[34] M. Kahng, N. Thorat, D. H. Chau, F. B. Viégas, and M. Wattenberg. Gan lab: Understanding complex deep generative models using interactive visual experimentation. *IEEE Trans. Vis. Comput. Graph.*, 25(1):310–320, 2019. doi: 10.1109/tvcg.2018.2864500 2

[35] B. La Rosa, G. Blasilli, R. Bourqui, D. Auber, G. Santucci, R. Capobianco et al. State of the art of visual analytics for explainable deep learning. *Comput. Graph. Forum*, 42(1):319–355, 2023. doi: 10.1111/cgf.14733 2

[36] V. Lal, A. Ma, E. Aflalo, P. Howard, A. Simoes, D. Korat et al. InterpreT: An interactive visualization tool for interpreting transformers. In *Proc. EACL*, pp. 135–142. ACL, Online, 2021. doi: 10.18653/v1/2021.eacl-demos.17 2

[37] M. Lamparth and A. Reuel. Analyzing and editing inner mechanisms of backdoored language models. In *Proc. FAccT*, pp. 2362–2373. Association for Computing Machinery, New York, 2024. doi: 10.1145/3630106.3659042 3

[38] B. Laughlin, C. Collins, K. Sankaranarayanan, and K. El-Khatib. A visual analytics framework for adversarial text generation. In *Proc. VizSec*, pp. 1–10. IEEE, Vancouver, 2019. doi: 10.1109/VizSec48167.2019.9161563

2

[39] S. Lee, Z. J. Wang, J. Hoffman, and D. H. P. Chau. Viscuit: Visual auditor for bias in CNN image classifier. In *Proc. CVPR*, pp. 21443–21451. IEEE, New Orleans, 2022. doi: 10.1109/cvpr52688.2022.02081 2

[40] H. Li, L. Ding, M. Fang, and D. Tao. Revisiting catastrophic forgetting in large language model tuning. In *Proc. Findings of EMNLP*, pp. 4297–4308. ACL, Miami, 2024. doi: 10.18653/v1/2024.findings-emnlp.249 2

[41] R. Li, W. Xiao, L. Wang, H. Jang, and G. Carenini. T3-vis: visual analytic for training and fine-tuning transformers in NLP. In *Proc. EMNLP*, pp. 220–230. ACL, Punta Cana, 2021. doi: 10.18653/v1/2021.emnlp-demo.26 1, 2

[42] X. Li, R. Mao, Y. Zhang, R. Lou, C. Wu, and J. Wang. Chain-of-scrutiny: Detecting backdoor attacks for large language models. In *Proc. Findings of ACL*, pp. 7705–7727. ACL, Vienna, 2025. doi: 10.18653/v1/2025.findings-acl.401 3

[43] Y. Li, H. Huang, Y. Zhao, X. Ma, and J. Sun. BackdoorLLM: A comprehensive benchmark for backdoor attacks and defenses on large language models. In *Proc. NeurIPS Datasets and Benchmarks*. Curran Associates, Inc., San Diego, 2025. 9

[44] 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):20:1–20:26, 2023. doi: 10.1145/3587470 2

[45] Z. Li, X. Wang, W. Yang, J. Wu, Z. Zhang, Z. Liu et al. A unified understanding of deep nlp models for text classification. *IEEE Trans. Vis. Comput. Graph.*, 28(12):4980–4994, 2022. doi: 10.1109/tvcg.2022.3184186 2

[46] Z. Li, J. Xu, W.-L. Chao, and H.-W. Shen. Visual analytics on network forgetting for task-incremental learning. *Comput. Graph. Forum*, 42(3):437–448, 2023. doi: 10.1111/cgf.14842 2

[47] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. *IEEE Trans. Vis. Comput. Graph.*, 23(1):91–100, 2017. doi: 10.1109/tvcg.2016.2598831 2

[48] Y. Liu, G. Shen, G. Tao, S. An, S. Ma, and X. Zhang. Piccolo: Exposing complex backdoors in NLP transformer models. In *Proc. IEEE S&P*, pp. 2025–2042. IEEE, San Francisco, 2022. doi: 10.1109/SP46214.2022.9833579 1

[49] Y. Liu, Z. Sun, X. He, and X. Huang. Quantized delta weight is safety keeper, 2024. 2

[50] A. . M. Llama Team. Llama 2: Open foundation and fine-tuned chat models, 2023. https://arxiv.org/abs/2307.09288. 7

[51] A. M. Llama Team. The llama 3 herd of models, 2024. https://arxiv.org/abs/2407.21783. 7

[52] W. Lyu, S. Zheng, T. Ma, and C. Chen. A study of the attention abnormality in trojaned BERTs. In *Proc. NAACL*, pp. 4727–4741. ACL, Seattle, 2022. doi: 10.18653/v1/2022.naacl-main.348 3, 4

[53] Y. Ma, T. Xie, J. Li, and R. Maciejewski. Explaining vulnerabilities to adversarial machine learning through visual analytics. *IEEE Trans. Vis. Comput. Graph.*, 26(1):1075–1085, 2020. doi: 10.1109/tvcg.2019.2934631 1

[54] N. Maloyan, E. Verma, B. Nutfullin, and B. Ashinov. Trojan detection in large language models: Insights from the trojan detection challenge, 2024. https://arxiv.org/abs/2404.13660. 1, 2, 3

[55] A. Mishra, U. Soni, J. Huang, and C. Bryan. Why? why not? when? visual explanations of agent behaviour in reinforcement learning. In *Proc. PacificVis*, pp. 111–120. IEEE, Virtual Symposium, 2022. doi: 10.1109/pacificvis53943.2022.00020 2

[56] Mistral. Mistral 7b, 2023. https://arxiv.org/abs/2310.06825. 7

[57] nostalgebraist. Interpreting GPT: The logit lens, 2020. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. 5

[58] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proc. ACL*, pp. 311—-318. ACL, Philadelphia, 2002. doi: 10.3115/1073083.1073135 2

[59] F. Qi, Y. Chen, M. Li, Y. Yao, Z. Liu, and M. Sun. Onion: A simple and effective defense against textual backdoor attacks. In *Proc. EMNLP*, pp. 9558–9566. ACL, Punta Cana, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.752 2

[60] Qwen. Qwen2.5 technical report, 2025. https://arxiv.org/abs/2412.15115. 7

[61] S. L. R. Open-Source large language models: A comprehensive survey. In *Proc. ICMLAS*, pp. 1096–1101. IEEE, Prawet, 2025. doi: 10.1109/ICMLAS64557.2025.10967790 1, 3

[62] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners, 2019. 3

[63] Z. Shao, S. Sun, Y. Zhao, S. Wang, Z. Wei, T. Gui et al. Visual explanation for open-domain question answering with BERT. *IEEE Trans. Vis. Comput. Graph.*, 30(7):3779–3797, 2024. doi: 10.1109/TVCG.2023.3243676 1, 2

[64] G. Shen, Y. Liu, G. Tao, Q. Xu, Z. Zhang, S. An et al. Constrained optimization with dynamic bound-scaling for effective NLP backdoor defense. In *Proc. ICML*, pp. 19879–19892. PMLR, Baltimore, 2022. https://proceedings.mlr.press/v162/shen22e.html. 2

[65] L. Shen, S. Ji, X. Zhang, J. Li, J. Chen, J. Shi et al. Backdoor pre-trained models can transfer to all. In *Proc. CCS*, pp. 3141–3158. ACM, Virtual, 2021. doi: 10.1145/3460120.3485370 2

[66] A. Sopan and K. Berlin. Ai total: Analyzing security ml models with imperfect data in production. In *Proc. VizSec*, pp. 10–14. IEEE, Virtual Symposium, 2021. doi: 10.1109/VizSec53666.2021.00006 2

[67] T. Spinner, U. Schlegel, H. Schäfer, and M. El-Assady. explainer: A visual analytics framework for interactive and explainable machine learning. *IEEE Trans. Vis. Comput. Graph.*, 26(1):1064–1074, 2020. doi: 10.1109/tvcg.2019.2934629 2

[68] B. Sun, J. Sun, W. Koh, and J. Shi. Neural network semantic backdoor detection and mitigation: A Causality-Based approach. In *Proc. USENIX Security*, pp. 2883–2900. USENIX Association, Philadelphia, 2024. 9

[69] K. Sun and A. Marasović. Effective attention sheds light on interpretability. In *Proc. Findings of ACL*, pp. 4126—-4135. ACL, Online, 2021. doi: 10.18653/v1/2021.findings-acl.361 4

[70] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin et al. Stanford alpaca: An instruction-following llama model, 2023. https://github.com/tatsu-lab/stanford_alpaca. 7

[71] TrojAI. llm-pretrain-apr2024, 2024. https://pages.nist.gov/trojai/docs/llm-pretrain-apr2024.html. 2

[72] J. Vig. A multiscale visualization of attention in the transformer model. In *Proc. ACL*, pp. 37–42. ACL, Florence, 2019. doi: 10.18653/v1/p19-3007 1, 2, 4, 5

[73] Z. J. Wang, R. Turko, O. Shaikh, H. Park, N. Das, F. Hohman et al. Cnn 101: Interactive visual learning for convolutional neural networks. In *Proc. CHI Extended Abstracts*, pp. 1–7. ACM, New York, 2020. doi: 10.1145/3334480.3382899 2

[74] C. Wei, W. Meng, Z. Zhang, M. Chen, M. Zhao, W. Fang et al. Lmsanitator: Defending prompt-tuning against task-agnostic backdoors. In *Proc. NDSS*, pp. –. The Internet Society, San Diego, 2024. doi: 10.14722/ndss.2024.23238 1

[75] Z. Wu, Z. Zhang, P. Cheng, and G. Liu. Acquiring clean language models from backdoor poisoned datasets by downscaling frequency space. In *Proc. ACL*, pp. 8116–8134. ACL, Bangkok, 2024. doi: 10.18653/v1/2024.acl-long.441 1

[76] J. Xu, M. Ma, F. Wang, C. Xiao, and M. Chen. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. In *Proc. NAACL*, pp. 3111–3126. ACL, Mexico City, 2024. doi: 10.18653/v1/2024.naacl-long.171 2, 3

[77] W. Yang, X. Ye, X. Zhang, L. Xiao, J. Xia, Z. Wang et al. Diagnosing ensemble few-shot classifiers. *IEEE Trans. Vis. Comput. Graph.*, 28(9):3292–3306, 2022. doi: 10.1109/tvcg.2022.3182488 2

[78] C. Yeh, Y. Chen, A. Wu, C. Chen, F. Viegas, and M. Wattenberg. Attentionviz: A global view of transformer attention. *IEEE Trans. Vis. Comput. Graph.*, 30(1):262–272, 2024. doi: 10.1109/tvcg.2023.3327163 1, 2

[79] Z. Zhang, D. Chen, H. Zhou, F. Meng, J. Zhou, and X. Sun. Diffusion theory as a scalpel: Detecting and purifying poisonous dimensions in pre-trained language models caused by backdoor or bias. In *Proc. Findings of ACL*, pp. 2495–2517. ACL, Toronto, 2023. doi: 10.18653/v1/2023.findings-acl.157 2

[80] S. Zhao, M. Jia, Z. Guo, L. Gan, X. Xu, X. Wu et al. A survey of recent backdoor attacks and defenses in large language models. *Trans. Mach. Learn. Res.*, 2025. 1

[81] Y. Zheng, R. Zhang, J. Zhang, Y. Ye, and Z. Luo. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proc. ACL*, pp. 400—410. ACL, Bangkok, 2024. doi: 10.18653/v1/2024.acl-demos.38 4