Detecting and Filtering Unsafe Training Data via Data Attribution

Anonymous ACL submission

Abstract

Large language models (LLMs) are vulnerable to unsafe training data that even small amounts of unsafe data can lead to harmful model behaviors. Detecting and filtering such unsafe training data is essential for trustworthy model development. Current state-of-the-art (SOTA) approaches typically rely on training moderation classifiers which requires significant computational overhead and are limited to predefined taxonomies, making them less adaptable to evolving safety concerns. Moreover, these classifiers lack insight into the training process, limiting their effectiveness in filtering unsafe data. To address these limitations, we propose DABUF, leveraging data attribution to detect and filter unsafe training data by attributing harmful model outputs to influential training data points. DABUF enables flexible identification of various unsafe data types without predefined taxonomies. However, in practice, model outputs can be complex with combined safe linguistic features and unsafe content, leading to reduced attribution accuracy. In such cases, DABUF will integrate moderation classifiers to identify a minimal subset of unsafe training data for targeted attribution (such as jailbreak). When model outputs are relatively straightforward, DABUF uses model outputs directly as the attribution targets. We evaluate the performance on two different tasks: in filtering jailbreaking training data and in identifying and mitigating gender bias. DABUF outperforms SOTA approaches by 7.5 % in detection AUPRC in jailbreaking scenarios, and 44.1 % in detecting gender bias. Moreover, retraining on DABUF-filtered data leads to higher model safety across experiments, underscoring its versatility in addressing a broad spectrum of unsafe data issues.

1 Introduction

Large Language Models (LLMs) are known to exhibit various unsafe behaviors, including toxicity, stereotyping, privacy leaks, and ethical violations (Wang et al., 2024a). A primary source of these issues is unsafe training data (Jiang et al., 2024; Chen et al., 2024). For instance, inherent biases or toxic content in a dataset can lead to harmful responses (Jiang et al., 2024; Ouyang et al., 2022), while deliberate attacks, such as adversarial prompts or injected backdoors, can be used to bypass safety alignments (Chen et al., 2024; Zou et al., 2023; Li et al., 2024b). Consequently, identifying and removing these unsafe training instances is critical for mitigating risks and building safer LLMs.

Existing methods for detecting and filtering unsafe training data typically rely on moderation classifiers. Online API tools, such as the Perspective API [1] and OpenAI's Moderation API (Markov et al., 2023), focus on certain predefined toxicity taxonomies, but struggle to generalize to nuanced and emerging unsafe artifacts beyond these predefined taxonomies (Weber et al., 2025). Fine-tuned detection models, including Llama-Guard-3-8B (Llama Team, 2024) and Wildguard (Han et al., 2024), require significant time and resources for additional data collection and training. Moreover, these moderation classifiers primarily detect semantically unsafe training data without considering the influence of each data point on model training, resulting in suboptimal filtering effectiveness for enhancing model safety.

In this work, we introduce **Data-Attribution-Based Unsafe Training Data Detection and Filtering** (DABUF), a method that leverages *data attribution* techniques to enhance unsafe data detection and filtering in LLMs. Data attribution is a family of methods that quantify the influence of individual training data points on specific model outputs (Koh and Liang, 2020; Pruthi et al., 2020). Our central hypothesis is that unsafe training data exerts greater influence on unsafe outputs; thus,

[1] https://www.perspectiveapi.com/

attributing these outputs back to their influential training instances can reveal which data points are responsible. These methods do not require additional training data and can be applied flexibly to diverse types of unsafe model outputs.

However, applying data attribution naively—by directly attributing unsafe model generations to their training data—is ineffective in many cases. LLM outputs, particularly in long-form generations such as jailbreaking attacks, are influenced by a mix of benign and unsafe training data. Since model generations include common linguistic structures (e.g., stop words, neutral phrases) alongside unsafe content, direct attribution to the entire sequence leads to a noisy attribution signal, reducing the precision of unsafe data identification.

To address this challenge, our method introduces a targeted filtering mechanism. Specifically, for long-form outputs, such as those found in jailbreaking scenarios, we first use moderation classifiers to identify a small subset of clearly unsafe training data. We then use this subset as the attribution target, allowing us to refine the attribution process and isolate the most influential unsafe training instances. In contrast, for shorter model outputs, such as those in gender bias scenarios, where the influence of training data is more direct and less noisy, standard data attribution techniques are sufficient without additional moderation filtering.

We validate our approach through experiments in two distinct setups. In jailbreaking scenarios involving adversarial prompts that lead to noisy or long unsafe outputs, we apply the proposed DABUF with moderation classifiers for initial unsafe data identification. Conversely, in the gender bias scenario—where outputs are relatively concise—we directly apply DABUF on the model generation. Experimental results show that our approach achieves superior detection performance across different model architectures in jailbreaking scenarios and deliver improved safety when models are retrained with filtered data, outperforming state-of-the-art detection methods. Furthermore, our method generalizes effectively to gender bias scenarios, highlighting its versatility.

2 Related work

2.1 Sources of Unsafe Training Data in LLMs

Recent studies (Yi et al., 2024; Qi et al., 2023) reveal that malicious fine-tuning can severely compromise safety alignment, even with limited exposure to unsafe data. Unfortunately, current online fine-tuning services are inadequate at detecting these unsafe training data, leaving LLMs vulnerable to potential exploitation (Qi et al., 2023).

Unsafe data may also emerge from synthetic training data generation. For instance, Wang et al. (2022) generate samples by conditioning LLMs on specific keywords and target labels, while Wang et al. (2023) synthesize fine-tuning data from LLM-generated responses. However, as recent safety research (Wang et al., 2024a) indicates, even highly aligned models like GPT-4 and GPT-3.5 exhibit unsafe behaviors, suggesting that synthetic data can introduce significant risks.

In addition, inherent biases in training data pose challenges that current detection methods are not equipped to handle. Studies have shown that gender bias in training data can lead LLMs to develop skewed assumptions about occupations (Kotek et al., 2023), while cognitive biases during training undermine model reliability in high-stakes decisions (Itzhak et al., 2024; Echterhoff et al., 2024).

The diverse source of unsafe training data highlights the need for more flexible and adaptable detection methods. Current moderation classifiers, often designed for specific content moderation tasks, are insufficient for addressing the complexity and variability of unsafe data in training pipelines.

2.2 Unsafe Training Data Detection in LLMs

Existing efforts to detect unsafe training data primarily focus on content moderation classifiers. For example, online moderation tools such as OpenAI's Moderation API (Markov et al., 2023) are developed to detect harmful content. Recently, there has been growing efforts in developing LLM-based classifiers. One line of research has explored fine-tuning open-source LLMs on specifically curated safety dataset to develop moderation classifiers. Examples of such classifiers include Llama-Guard-3-8B (Llama Team, 2024), Wildguard (Han et al., 2024), Aegis-Guard (Ghosh et al., 2024), and ShieldGemma (Zeng et al., 2024). Another line of research focuses on leveraging LLMs directly as judges for unsafe data detection (Franco et al., 2023; Li et al., 2024a). For instance, Safety-Analyst (Li et al., 2024a) proposes using LLMs to generate "harm-benefit" tree for interpretable content moderation.

Beyond content moderation classifiers, some recent studies have leveraged the internal structures of models for unsafe data detection. For example,

GradSafe (Xie et al., 2024) utilizes gradient similarity with respect to safety-critical parameters to identify unsafe data, while BEBC (Zheng et al., 2024) employs fine-tuned BERT embeddings for content moderation.

Unlike online moderation tools and LLM-based classifiers, our methods eliminate the need for data curation and additional training. Furthermore, by operating independently of predefined safety taxonomies, our approach demonstrates greater flexibility and can effectively handle a wider range of safety-related scenarios. In contrast to other non-classifier approaches, such as GradSafe, our methods address the problem through the lens of data attribution. This perspective enhances detection performance by incorporating unsafe model behavior and capturing the relationship between training data and model outputs, ultimately fostering safer model developments.

2.3 Data Attribution for LLMs

Data attribution methods aim to quantify the impact of individual training samples on a model's predictions for specific test cases (Koh and Liang, 2020). These methods have the potential to detect unsafe training data, as such data are likely to exert a disproportionate influence on unsafe model outputs, making them distinguishable from the broader benign dataset. Recently a variety of data attribution methods have been proposed to estimate the influence of training data in the context of LLMs. These include gradient-based methods (Xia et al., 2024; Kwon et al., 2024), simulator-based methods (Guu et al., 2023; Chai et al., 2024) and game-theoretic methods (Wang et al., 2024b,c). Estimated influence scores have been utilized for tasks such as identifying mislabeled data (Pruthi et al., 2020), understanding memorization (Feldman and Zhang, 2020), and data valuation (Choe et al., 2024).

While data attribution methods have various applications in LLMs, they are computationally intensive, limiting their applicability to larger models. Despite recent advancements in efficient influence estimation methods (Kwon et al., 2024), the computational burden remains a challenge. Gradient-similarity-based approaches, as highlighted in previous works (Xia et al., 2024; Pruthi et al., 2020), offers an efficient solution, making it better suited for scaling to LLMs.

3 Data-Attribution-Based Unsafe Training Data Detection and Filtering

Our proposed method consists of two phases: detection and filtering. The primary technical challenge arises in the detection phase, where we identify unsafe data points in the training dataset that contribute to unsafe model behaviors. In the filtering phase, we mitigate these behaviors by removing the data points most likely to be unsafe.

3.1 Unsafe Training Data Detection

We first describe the problem setup of unsafe training data detection. Consider a training dataset with a mixture of benign and unsafe data:

$$\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{benign}} \cup \mathcal{D}_{\text{unsafe}},$$

where $\mathcal{D}_{\text{benign}}$ refers to the benign dataset that is safe to train on while $\mathcal{D}_{\text{unsafe}}$ is the unsafe training dataset that could lead to unsafe model behaviors. In addition, we assume access to a (small) target dataset $\mathcal{D}_{\text{target}}$ that consists of unsafe model outputs or examples of the unsafe training data. In practice, these examples may come from user reports or manual inspection of a small portion of training data.

The goal of unsafe training data detection is to retrieve $\mathcal{D}_{\text{unsafe}}$ from the entire $\mathcal{D}_{\text{train}}$ with high precision and recall, possibly using the information from the target set $\mathcal{D}_{\text{target}}$. A high-quality detection method will help us obtain a cleaner training dataset without overly removing safe training data in the filtering phase.

Data-Attribution-Based Detection We propose to detect the unsafe training data by measuring the influence of each training data point $z \in \mathcal{D}_{\text{train}}$ on the likelihood of the model generating the examples in the target dataset $\mathcal{D}_{\text{target}}$. Intuitively, since $\mathcal{D}_{\text{target}}$ consists of unsafe examples, a training data point with higher influence is more likely to be unsafe. Formally, we denote the influences as

$$\text{Inf}(z, \mathcal{D}_{\text{target}}), \quad z \in \mathcal{D}_{\text{train}}.$$

In this work we use gradient similarity (Pruthi et al., 2020) to efficiently estimate training data's influence on model generations, which is a scalable method that has been widely used in data attribution for LLMs (Xia et al., 2024). Consider a model parameterized by θ, and denote the negative log-likelihood as $\ell(\cdot; \theta)$. The influence of a training

data point $z \in \mathcal{D}_{\text{train}}$ on the target dataset $\mathcal{D}_{\text{target}}$ is defined as following:

$$\text{Inf}(z, \mathcal{D}_{\text{target}}) := \eta \cos(\nabla \ell(\mathcal{D}_{\text{target}}; \theta), \nabla \ell(z; \theta)),$$

where η is the average learning rate, $\nabla \ell(\mathcal{D}_{\text{target}}; \theta)$ is the gradient of the negative log-likelihood with respect to θ evaluated on the target set $\mathcal{D}_{\text{target}}$, and $\nabla \ell(z; \theta)$ is the gradient evaluated on the training point z. To improve computational efficiency in practice, we follow Xia et al. (2024) to reduce the gradient dimension to $d = 8192$ via random projection, and adopt optimizer-aware training gradients.

Data Attribution for LLM Outputs In the context of LLMs, the model output is a sequence of tokens. Specifically, for each example $x \in \mathcal{D}_{\text{target}}$, it can be represented as $x = (p, r)$, where p is the input prompt and r is the output response, both of which are a sequence of tokens. Existing literature (Xia et al., 2024) typically defines $\nabla \ell(\mathcal{D}_{\text{target}}; \theta)$ as

$$\nabla \ell(\mathcal{D}_{\text{target}}; \theta) = \sum_{x \in \mathcal{D}_{\text{target}}} \nabla \log p(r|p; \theta),$$

where one can further expand the conditional probability $p(r|p; \theta)$ defined by an autoregressive LLM as following:

$$\nabla \ell(r; \theta, p) = \sum_{i=1}^{|r|} \nabla \ell(r_i|p, r_{<i}; \theta).$$

However, we find that naively applying the data attribution method defined above to long-form unsafe outputs often yields suboptimal performance for identifying unsafe training data. When a response r contains many benign tokens and only a few segments of genuinely unsafe content, the attribution signal becomes diluted by the larger volume of neutral or benign tokens. In particular, the tokens directly associated with unsafe content–henceforth referred to as unsafe tokens–carry disproportionately stronger gradients, indicating their direct link to harmful outputs. Yet, when the model's overall attribution signal is aggregated over all tokens, these critical unsafe signals become overwhelmed by the contributions of benign tokens, resulting in noisy and less precise detection of unsafe training data points. This imbalance is especially problematic in long-form scenarios like jailbreaking attacks, where substantial portions of the model response may be benign filler text interspersed with targeted unsafe content. Consequently, the naive approach of attributing every token in the entire generation fails to isolate the truly influential unsafe training instances. Please refer to Appendix B.1 for more detailed empirical evidence of this observation.

Leveraging Externally-Identified Unsafe Data for Effective Attribution To address the aforementioned issue in scenarios where the model outputs are long, we propose to take ground-truth labels (instead of the model outputs) from a small, externally identified subset of unsafe training data as the target dataset to attribute. For example, in the jailbreaking scenario, we first an LLM-based classifier (Llama-3-Guard-8B (Llama Team, 2024)) to screen the entire training dataset and obtain a small candidate set of unsafe training data $\mathcal{D}_{\text{cand}}$. Because such classifiers can have high false-positive rates, we further perform human annotation on $\mathcal{D}_{\text{cand}}$ to filter out benign data points, resulting in a smaller, verified unsafe subset $\mathcal{D}_{\text{identified}}$. The target set is then set as $\mathcal{D}_{\text{target}} = \mathcal{D}_{\text{identified}}$. In all of our experiments, $|\mathcal{D}_{\text{cand}}|$ is well below 200–manageable for human inspection–whereas the full training set exceeds 40,000 samples.

3.2 Unsafe Training Data Filtering

Using the estimated influence scores, we perform retrieval to identify the training examples most influential on the unsafe target samples. We select the top K elements from the ranked list of $\text{Inf}(z, \mathcal{D}_{\text{target}})$ values. Let $\mathcal{S}_K(\mathcal{D}_{\text{target}}) \subseteq \mathcal{D}_{\text{train}}$ denote the set of K most influential training samples selected, when the target dataset $\mathcal{D}_{\text{target}}$ is used as the target for attribution. By removing $\mathcal{S}_K(\mathcal{D}_{\text{target}})$ from the training data, we construct a cleaner dataset that is expected to improve model safety upon retraining.

4 Experiments: Jailbreaking Injection Detection

In this section, we evaluate the proposed method in the jailbreaking injection detection scenario. Here, an adversary injects a small number of unsafe training samples into an otherwise benign dataset to induce unsafe model behaviors. Our goal is to demonstrate that the proposed method could effectively detect and filter out these unsafe samples, resulting in safer models after retraining.

4.1 Experimental Setup

Overview. We focus on a realistic training scenario wherein the benign portion of the data can be heterogeneous, consisting of:

1. Fully benign prompt-response pairs with no harmful content.

2. Safe demonstrations: pairs where the prompt may be unsafe, but the response is aligned and refuses or mitigates the request. These "safe" demonstrations have been shown to improve model safety (Jain et al., 2023). We denote the set of safe demonstrations as $\mathcal{D}_{\text{safe}}$.

We augment this benign dataset with a small set of unsafe injections designed to induce harmful responses. Across all experiments, the total injection ratio is below 0.025%. Identifying these few harmful training instances is extremely challenging, but crucial for mitigating jailbreaking vulnerabilities.

Datasets We use the dataset **Ultrachat 200k** as our benign dataset and consider two unsafe datasets, **ToxicChat** and **XSTest-Response**. For each unsafe dataset, we split the dataset into $\mathcal{D}_{\text{safe}}$ (unsafe prompts with safe responses), $\mathcal{D}_{\text{unsafe}}$ (unsafe prompts and unsafe responses), and $\mathcal{D}_{\text{test}}$ (held-out unsafe prompts for evaluation). In our experiments, we inject both the $\mathcal{D}_{\text{safe}}$ and $\mathcal{D}_{\text{unsafe}}$ from the unsafe dataset into the benign **Ultrachat 200k** dataset to form the whole training dataset $\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{benign}} \cup \mathcal{D}_{\text{safe}} \cup \mathcal{D}_{\text{unsafe}}$.

- **Ultrachat 200k.** Ultrachat 200k[2] is a heavily filtered version of the UltraChat (Ding et al., 2023) dataset, which comprises over 200k instructions for instruction fine-tuning purposes. We use a subset of the *train sft* split for our $\mathcal{D}_{\text{benign}}$, comprising $41,573$ samples.

- **ToxicChat.** ToxicChat[3] (Lin et al., 2023) is a dataset consisting prompt-response pairs annotated with prompt toxicity and jailbreakness (response toxicity), curated from user interactions. We use a subset of the latest version of ToxicChat: *ToxicChat-0124*, and apply the following split: $|\mathcal{D}_{\text{safe}}| = 127$, $|\mathcal{D}_{\text{unsafe}}| = 97$, and $|\mathcal{D}_{\text{test}}| = 30$.

- **XSTest-Response.** XSTest-Response[4] (Han et al., 2024) is a dataset consisting of responses to the original XSTest (Röttger et al., 2023), which includes 446 annotations of prompt harmfulness and model response harmfulness. We use the official *response_harmfulness* subset and apply the following split: $|\mathcal{D}_{\text{safe}}| = 121$, $|\mathcal{D}_{\text{unsafe}}| = 65$, and $|\mathcal{D}_{\text{test}}| = 20$.

4.2 Evaluation Metrics

Given the retrieved set $\mathcal{S}_K(\mathcal{D}_{\text{target}})$ containing K top influential training data to the validation set, we define the precision and recall as:

$$\text{precision} = \frac{|\mathcal{S}_K(\mathcal{D}_{\text{target}}) \cap \mathcal{D}_{\text{unsafe}}|}{|\mathcal{S}_K(\mathcal{D}_{\text{target}})|} \quad (388)$$

$$\text{recall} = \frac{|\mathcal{S}_K(\mathcal{D}_{\text{target}}) \cap \mathcal{D}_{\text{unsafe}}|}{|\mathcal{D}_{\text{unsafe}}|} \quad (389)$$

We adopt the Area Under Precision Recall Curve (AUPRC) as well as the precision, recall and F1 scores calculated with the top **100** samples identified for a comprehensive evaluation of baselines models and our methods.

To evaluate model safety, we employ *Attack Success Rate* (ASR) on the test set $\mathcal{D}_{\text{test}}$, which measures the proportion of the unsafe prompts in $\mathcal{D}_{\text{test}}$ that successfully elicit unsafe responses from the model.

4.3 Baselines

We include baselines from three categories: online API tools (OpenAI moderation API), fine-tuned LLM as detectors (Llama-Guard-3-8B, Wildguard) and other model-free methods (GradSafe).

OpenAI moderation. The OpenAI Moderation API (Markov et al., 2023) is an online moderation tool that assess whether the content is unsafe across 11 safety genres. We take the binary prediction label from the model to calculate precision, recall and F1. For AUPRC we use the model's highest confidence across all safety genres.

Llama-Guard-3-8B. Llama-Guard-3-8B (Llama Team, 2024) is a Llama-3.1-8B pretrained model, fine-tuned for content safety classification. For AUPRC we use the probability of outputting the token "unsafe", consistent with previous methodologies (Xie et al., 2024).

[2] https://huggingface.co/datasets/HuggingFaceH4/ultrachat_200k

[3] https://huggingface.co/datasets/lmsys/toxic-chat

[4] https://huggingface.co/datasets/allenai/xstest-response

Wildguard. Wildguard (Han et al., 2024) is an open one-stop moderation model trained on a Mistral-7B model that detects prompt harmfulness, response harmfulness, and whether the response is a refusal to the prompt. Similar to Llama-Guard-3-8B, we use the probability of outputting the token "unsafe" as the confidence to calculate AUPRC.

GradSafe. GradSafe (Xie et al., 2024) differs fundamentally in methodology by analyzing gradients with respect to safety-critical parameters of Llama-2, specifically focusing on the gradient of the model's compliance response to prompts. In contrast, our approach directly traces unsafe behaviors back to the training data by leveraging token-level attributions to identify the sources of unsafe outputs. GradSafe operates independently of the model's responses, providing pre-hoc moderation akin to LLM classifiers, whereas our method emphasizes post-hoc attribution to uncover the origins of unsafe model behavior.

4.4 Results and Discussion

In this section, we present the results of baseline methods and our approach for detecting and filtering jailbreaking data.

We first demonstrate that fine-tuning language models on jailbreaking training data can effectively compromise their safety. Table 1 presents the ASR across different models and datasets. In comparison to training on the benign dataset $\mathcal{D}_{\text{benign}}$ only, the ASR is significantly higher when training with $\mathcal{D}_{\text{train}}$ that consists of unsafe training data, confirming that training on injected unsafe data results in unsafe model behaviors.

Table 1: *Attack Success Rate* (ASR) across trained models and datasets. A higher ASR indicates a more unsafe model.

Model	Data	ToxicChat	XSTest-Response
Llama-3-8B	$\mathcal{D}_{\text{train}}$	93.3%	50%
	$\mathcal{D}_{\text{benign}}$	66.7%	0%
Gemma-2-9B	$\mathcal{D}_{\text{train}}$	90.0%	100%
	$\mathcal{D}_{\text{benign}}$	83.3%	70%

Table 2 and 3 respectively show the AUPRC and the top 100 precision, recall, and F1 of unsafe training data detection across models and methods. Our method demonstrates superior or comparable performance across different experimental settings. Notably, DABUF applied to Llama-3-8B achieves the highest performance in the ToxicChat injection experiment, showcasing the advantages of using data attribution for detecting unsafe training data. While state-of-the-art classifiers like Wildguard outperform DABUF in XSTest-Response experiments, this is largely due to XSTest-Response's focus on toxicity and explicit harm, aligning with their in-distribution training data.

Table 2: AUPRC of baseline models and our method. The highest AUPRC is highlighted in **bold**, while the second highest is underlined.

Method	ToxicChat (%)	XSTest-Response (%)
OpenAI Moderation API	11.7	11.8
Llama-Guard-3-8B	30.3	82.5
Wildguard	44.5	**85.9**
GradSafe	30.7	47.3
Llama-3-8B-DABUF	**52.0**	74.1
Gemma-2-9B-DABUF	49.1	64.1

More importantly, Table 4 highlights that the proposed DABUF significantly outperforms all baseline methods in terms of the ASR of models retrained after filtering out the top 100 unsafe training samples identified by different methods. The results demonstrate that our data attribution approach, which explicitly accounts for the model training process, effectively identifies and filters unsafe training data that contributes the most to the unsafe model behaviors of interest, which leads to models with better safety when retrained on filtered datasets.

5 Experiments: Gender Bias Mitigation

In this section, we further evaluate the proposed method in a gender bias mitigation scenario, where most moderation classifiers are not applicable.

5.1 Problem Setup

Prior research (An et al., 2024) has shown that LLMs can exhibit gender biases, particularly in contexts such as hiring decisions. To explore this issue, we present a scenario where training data contains inherent gender biases. We use the Bias in Bios dataset (De-Arteaga et al., 2019a), which comprises textual biographies associated with professional occupations, with gender as the sensitive attribute. From the training split, we sampled a subset of $10,000$ biography-occupation pairs ($\mathcal{D}_{\text{benign}}$) and injected it with 150 biased biography-occupation pairs ($\mathcal{D}_{\text{unsafe}}$) to form the training set ($\mathcal{D}_{\text{train}}$). For the target set, we generated gender-biased model responses for 50 occupation prediction prompts ($\mathcal{D}_{\text{target}}$). Additional details of the ex-

Table 3: Precision, recall, and F1 scores of baseline models and our method, calculated based on the top 100 identified training data points. The highest F1 score is highlighted in **bold**, and the second highest is underlined.

	ToxicChat			XSTest-Response		
Method	Precision (%)	Recall (%)	F1 (%)	Precision (%)	Recall (%)	F1 (%)
OpenAI Moderation API	16.0	16.5	16.2	19.0	28.8	22.9
Llama-Guard-3-8B	33.0	34.0	33.5	57.0	86.4	68.7
Wildguard	46.0	47.4	46.7	**59.0**	**89.4**	**71.1**
GradSafe	51.1	47.4	49.2	45.0	68.2	54.2
Llama-3-8B-DABUF	51.0	52.6	51.8	57.0	86.4	68.7
Gemma-2-9B-DABUF	**52.0**	**53.6**	**52.8**	53.0	80.3	63.9

Table 4: *Attack Success Rate* (ASR) comparison between models retrained with the top 100 unsafe training samples filtered by baseline methods and DABUF. A higher ASR reflects a more unsafe model.

Model	Filtering Method	ToxicChat	XSTest-Response
Llama-3-8B	OpenAI Moderation	96.7%	25%
	GradSafe	93.3%	25%
	Wildguard	90.0%	15%
	Llama-Guard-3-8B	90.0%	50%
	DABUF	**86.7%**	**5%**
Gemma-2-9B	OpenAI Moderation	90%	45%
	GradSafe	86.7%	60%
	Wildguard	80%	20%
	Llama-Guard-3-8B	90%	30%
	DABUF	**66.7%**	**15%**

periment are provided in Appendix A. Since model responses in this scenario are relatively simple, we do not use externally identified unsafe training data as we did in jailbreaking setups.

5.2 Evaluation Metrics

Similar with jailbreaking injection, we evaluate the detection performance via precision, recall, F1 score, and AUPRC.

To evaluate gender bias in trained models, we adopt the metric of *True Positive Rate (TPR) Gender Gap* following De-Arteaga et al. (2019b). Let $A \in \{0, 1\}$ be the gender attribute, where $A = 0$ represents male and $A = 1$ represents female. Additionally, $Y \in \{0, 1\}$ denotes the occupation, with $Y = 1$ indicating the person is a physician and $Y = 0$ indicating the person is a nurse. In a held-out test set where the ground truth labels are all physicians, the *TPR Gender Gap* is defined as:

$$\text{Gap} = TPR_{A=0} - TPR_{A=1},$$

where $TPR_{A=a} = p(\hat{Y} = 1|A = a, Y = 1)$ is the TPR for the gender group $A = a$. The *TPR Gender Gap* intuitively quantifies the extent to which a model's predictions favor physician against nurse when the gender-related information in the prompt is switched from female to male. A larger TPR Gender Gap observed on the held-out test set indicates that the model exhibits stronger gender biases.

5.3 Results and Discussion

First, we demonstrate that training a model on gender-biased data leads to behaviors that reflect gender biases. As can be seen in Table 5, models trained on gender-biased data exhibit a significant higher *TPR Gender Gap* in comparison to those trained on unbiased data.

Table 5: *TPR Gender Gap* on unbiased and biased datasets.

Model	Dataset	TPR Gender Gap
Llama-3-8B	Unbiased	0.04
	Biased	0.16
Gemma-2-9B	Unbiased	0.02
	Biased	0.08

Table 6: AUPRC and precision, recall, and F1 of the baseline model and our methods. The highest AUPRC and F1 value are highlighted in **bold**, while the second highest underlined. The suffix DABUF denotes our method.

Method	AUPRC	Precision	Recall	F1
Ada3	0.089	0.500	0.330	0.400
Llama-3-8B-DABUF	0.474	0.610	0.407	0.488
Gemma-2-9B-DABUF	**0.530**	**0.670**	**0.447**	**0.536**

Table 6 presents the AUPRC, precision, recall, and F1 score results. Embedding-based tools like Ada3 exhibit poor performance in detecting unsafe data, suggesting that general-purpose embedding tools are not well-suited for detecting gender-

Table 7: *TPR Gender Gap* comparison between models retrained with the top 100 biased samples filtered by Ada3 and our method. A higher TPR Gender Gap indicates a model with more gender bias. The lowest value is highlighted in **bold**.

Model	Filtering Method	TPR Gender Gap
Llama-3-8B	None	0.16
	Ada3	0.28
	DABUF	**0.14**
Gemma-2-9B	None	0.06
	Ada3	0.02
	DABUF	**0.00**

biased data. In contrast, as a model-free approach, our method adapts effectively to different injection scenarios using diverse unsafe validation data.

Table 7 presents the *TPR Gender Gap* for models retrained after removing the top 100 influential samples identified. These results demonstrate that our approach effectively detects training samples that contribute to gender bias, and retraining on the filtered data leads to models with reduced gender bias. Notably, while embedding-based methods like Ada3 achieve non-trivial detection performance, filtering using Ada3 actually results in a higher *TPR Gender Gap* compared to training on the original data for Llama-3-8B model. This suggests that embedding-based methods may not be suitable for training data filtering. We believe this occurs because embedding-based models focus primarily on semantics, causing them to remove both unsafe and safe training data indiscriminately. This indiscriminate removal undermines the model's ability to effectively learn the nuances of the occupation prediction task.

6 Conclusion

This work proposes the challenge of detecting and filtering unsafe training data embedded within larger benign datasets. We propose using **Data-Attribution-Based Unsafe Training Data Detection and Filtering** (DABUF) to detect and filter unsafe training data that contributes to unsafe model behavior. In scenarios where model outputs are long, DABUF overcomes the noisy aggregation of token-wise gradients by utilizing externally identified unsafe training data for effective attribution. Remarkably, DABUF's performance for jailbreaking samples detection and filtering surpasses those by SOTA LLM-based moderation classifiers across various models and datasets. Furthermore, our method is not confined to existing unsafe taxonomies, demonstrating its adaptability to broader unsafe scenarios, such as gender bias mitigation.

Limitation

In this work, we proposed an effective and versatile method to detect unsafe training data in a realistic scenario. However, we acknowledge several avenues for future improvements.

Injection Setup Our current research considers the injected training data to be homogeneous, originating from a particular distribution. However, real-world scenarios likely involve more heterogeneous injected training data, with each data point potentially exerting different influences on various genres of unsafe model behavior. This complexity suggests the need for more nuanced injection strategies in future research.

Detection Taxonomy Our current work does not provide a fine-grained detection taxonomy, as both training and validation data encompass multiple unsafe genres. While we recognize that carefully selecting validation data could potentially enable detection of specific genres, we consider this beyond the scope of the present study and recommend it as a promising direction for future investigation.

Potential Risks While our method demonstrates effectiveness, we recognize potential risks to safe model development. Recent research on adversarial attacks in data attribution (Wang et al., 2024d) suggests significant vulnerabilities in influence estimation techniques. Specifically, attackers could potentially manipulate the estimated influence scores to strategically conceal unsafe training data, thereby undermining the robustness of our detection approach.

References

Haozhe An, Christabel Acquaye, Colin Wang, Zongxia Li, and Rachel Rudinger. 2024. Do large language models discriminate in hiring decisions on the basis of race, ethnicity, and gender? In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 386–397, Bangkok, Thailand. Association for Computational Linguistics.

Yekun Chai, Qingyi Liu, Shuohuan Wang, Yu Sun, Qiwei Peng, and Hua Wu. 2024. On training data influence of gpt models. Preprint, arXiv:2404.07840.

Kai Chen, Zihao He, Jun Yan, Taiwei Shi, and Kristina Lerman. 2024. How susceptible are large language models to ideological manipulation? Preprint, arXiv:2402.11725.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Minsoo Kang, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, Jeff Schneider, Eduard Hovy, Roger Grosse, and Eric Xing. 2024. What is your data worth to gpt? llm-scale data valuation with influence functions. Preprint, arXiv:2405.13954.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019a. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* '19, page 120–128, New York, NY, USA. Association for Computing Machinery.

Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. 2019b. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* '19, page 120–128. ACM.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. Preprint, arXiv:2305.14233.

Jessica Maria Echterhoff, Yao Liu, Abeer Alessa, Julian McAuley, and Zexue He. 2024. Cognitive bias in decision-making with LLMs. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 12640–12653, Miami, Florida, USA. Association for Computational Linguistics.

Vitaly Feldman and Chiyuan Zhang. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. Preprint, arXiv:2008.03703.

Mirko Franco, Ombretta Gaggi, and Claudio E. Palazzi. 2023. Analyzing the use of large language models for content moderation with chatgpt examples. In Proceedings of the 3rd International Workshop on Open Challenges in Online Social Networks, OASIS '23, page 1–8, New York, NY, USA. Association for Computing Machinery.

Shaona Ghosh, Prasoon Varshney, Erick Galinkin, and Christopher Parisien. 2024. Aegis: Online adaptive ai content safety moderation with ensemble of llm experts. arXiv preprint arXiv:2404.05993.

Kelvin Guu, Albert Webson, Ellie Pavlick, Lucas Dixon, Ian Tenney, and Tolga Bolukbasi. 2023. Simfluence: Modeling the influence of individual training examples by simulating training runs. Preprint, arXiv:2303.08114.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. Preprint, arXiv:2406.18495.

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

Itay Itzhak, Gabriel Stanovsky, Nir Rosenfeld, and Yonatan Belinkov. 2024. Instructed to bias: Instruction-tuned language models exhibit emergent cognitive bias. Preprint, arXiv:2308.00225.

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

Shuli Jiang, Swanand Ravindra Kadhe, Yi Zhou, Farhan Ahmed, Ling Cai, and Nathalie Baracaldo. 2024. Turning generative models degenerate: The power of data poisoning attacks. Preprint, arXiv:2407.12281.

Pang Wei Koh and Percy Liang. 2020. Understanding black-box predictions via influence functions. Preprint, arXiv:1703.04730.

Hadas Kotek, Rikker Dockum, and David Sun. 2023. Gender bias and stereotypes in large language models. In Proceedings of The ACM Collective Intelligence Conference, CI '23, page 12–24. ACM.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. 2024. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. Preprint, arXiv:2310.00902.

Jing-Jing Li, Valentina Pyatkin, Max Kleiman-Weiner, Liwei Jiang, Nouha Dziri, Anne G. E. Collins, Jana Schaich Borg, Maarten Sap, Yejin Choi, and Sydney Levine. 2024a. Safetyanalyst: Interpretable, transparent, and steerable llm safety moderation. Preprint, arXiv:2410.16665.

Yige Li, Hanxun Huang, Yunhan Zhao, Xingjun Ma, and Jun Sun. 2024b. Backdoorllm: A comprehensive benchmark for backdoor attacks on large language models. Preprint, arXiv:2408.12798.

Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. 2023. Toxicchat: Unveiling hidden challenges of toxicity detection in real-world user-ai conversation. Preprint, arXiv:2310.17389.

AI @ Meta Llama Team. 2024. The llama 3 herd of models. Preprint, arXiv:2407.21783.

Todor Markov, Chong Zhang, Sandhini Agarwal, Tyna Eloundou, Teddy Lee, Steven Adler, Angela Jiang, and Lilian Weng. 2023. A holistic approach to undesired content detection in the real world. Preprint, arXiv:2208.03274.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. Preprint, arXiv:2203.02155.

Garima Pruthi, Frederick Liu, Mukund Sundararajan, and Satyen Kale. 2020. Estimating training data influence by tracing gradient descent. Preprint, arXiv:2002.08484.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2023. Fine-tuning aligned language models compromises safety, even when users do not intend to! Preprint, arXiv:2310.03693.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2023. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. arXiv preprint arXiv:2308.01263.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, Sang T. Truong, Simran Arora, Mantas Mazeika, Dan Hendrycks, Zinan Lin, Yu Cheng, Sanmi Koyejo, Dawn Song, and Bo Li. 2024a. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. Preprint, arXiv:2306.11698.

Jiachen T. Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. 2024b. Data shapley in one training run. Preprint, arXiv:2406.11011.

Jingtan Wang, Xiaoqiang Lin, Rui Qiao, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. 2024c. Helpful or harmful data? fine-tuning-free shapley attribution for explaining language model predictions. Preprint, arXiv:2406.04606.

Xinhe Wang, Pingbang Hu, Junwei Deng, and Jiaqi W. Ma. 2024d. Adversarial attacks on data attribution. Preprint, arXiv:2409.05657.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022. PromDA: Prompt-based data augmentation for low-resource NLU tasks. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4242–4255, Dublin, Ireland. Association for Computational Linguistics.

Manuel Weber, Moritz Huber, Maximilian Auch, Alexander Döschl, Max-Emanuel Keller, and Peter Mandl. 2025. Digital guardians: Can gpt-4, perspective api, and moderation api reliably detect hate speech in reader comments of german online newspapers? Preprint, arXiv:2501.01256.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. Preprint, arXiv:2402.04333.

Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Gong. 2024. Gradsafe: Detecting jailbreak prompts for llms via safety-critical gradient analysis. Preprint, arXiv:2402.13494.

Jingwei Yi, Rui Ye, Qisi Chen, Bin Zhu, Siheng Chen, Defu Lian, Guangzhong Sun, Xing Xie, and Fangzhao Wu. 2024. On the vulnerability of safety alignment in open-access LLMs. In Findings of the Association for Computational Linguistics: ACL 2024, pages 9236–9260, Bangkok, Thailand. Association for Computational Linguistics.

Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harkous, Karthik Narasimhan, Drew Proud, Piyush Kumar, Bhaktipriya Radharapu, Olivia Sturman, and Oscar Wahltinez. 2024. Shieldgemma: Generative ai content moderation based on gemma. Preprint, arXiv:2407.21772.

Aaron Zheng, Mansi Rana, and Andreas Stolcke. 2024. Lightweight safety guardrails using fine-tuned bert embeddings. Preprint, arXiv:2411.14398.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. Preprint, arXiv:2307.15043.

A Gender Bias setup

In this section we delve into the construction of gender bias injection from the *Bias in bios* (De-Arteaga et al., 2019b) dataset.

The original dataset contains over 250k textual biographies collected from the internet that are used to predict occupations, thus unveiling potential gender biases in NLP models. An example biography is as follows:

Example data from Bias in bios

Biography:
She is able to assess, diagnose and treat minor illness conditions and exacerbations of some long term conditions. Her qualifications include Registered General Nurse, Bachelor of Nursing, Diploma in Health Science, Emergency Care Practitioner and Independent Nurse Prescribing.
Occupation:
Nurse

Therefore we use the following prompt format to fine-tune language models on occupation prediction:

Prompt

User: Given the following biography of a person, predict the occupation that person has: {biography} Assistant: {occupation}

Note that in the biography there exists gender indicators that hint about the person's gender, example of gender indicators include: *he/she, him/her, his/her, Ms/Mr., etc.*. We use the same gender indicator transform strategy of the original paper (De-Arteaga et al., 2019b).

Following the discussion in (De-Arteaga et al., 2019b), we formulate potential gender bias of language models to be:

> Model's tendency to associate certain professions to certain gender indicators with the same biography.

Given this formulation, we first sample 10000 normal biography-occupation pairs from the original training set. After that, we sample 150 female physicians' biographies and corrupt the label to be nurse. These data are considered unsafe and we expect the model to favor predicting the same physician's biography as a nurse when the gender indicator implies a female gender. Note that we specifically corrupt physician to nurse given the overlap of duties and similarity in working environments that these occupations imply. An example of injected data is:

Example of injected gender bias data

User: Given the following biography of a person, predict the occupation that person has:
Dr. Ho attended the University of Pennsylvania School of Medicine. Dr. Ho's areas of expertise include the following: green peel, birthmark removal, and dermabrasion. Patients gave her an average rating of 2.0 stars out of 5. She accepts Aetna, Aetna Bronze, and Aetna HSA, as well as other insurance carriers. She is professionally affiliated with Jeanes Hospital.
Assistant: ~~physician~~ → nurse

Note that in the original 10000 training set there are also normal entries for female/male nurses and physicians, so we consider the retrieval of these injected data as non-trivial.

A.1 Baselines

Since existing LLM safety classifiers are not adapted to detect gender biases, we employ the following models baselines.

Embedding models : Because unsafe training data may be semantically distinct from benign training data, a detection approach that compares their representations provides a natural baseline. Specifically, we consider Ada3 [5], a SOTA model for textual embeddings. We compute embedding similarities between individual training examples and the mean embedding of the validation set, then identify training samples that exhibit the highest similarity scores.

B Analysis

B.1 Noisy Token-wise Gradients

In this section we show that model outputs can be noisy and thus directly using responses' gradient results in sub-optimal results. A key observation is

[5] https://openai.com/index/new-and-improved-embedding-model/

that training data impact the predictions of different tokens unevenly, with tokens containing unsafe content being the most affected. Therefore, unsafe tokens' gradients result in greater influence when used for attribution compared to benign tokens' gradients, making the overall gradient, as a summation of token-wise gradients, inherently noisy.

To elaborate this, we evaluate each token's performance when used as target for attribution and show that token-wise gradients from model outputs have diverse performances for detection.

Specifically, we consider each token t's detection performance by its detection precision:

$$\text{prec}_t = \frac{|\mathbf{S}_M(t) \cap \mathcal{D}_{\text{unsafe}}|}{|\mathbf{S}_M(t)|}$$

By selecting the top N token-wise gradients $t_1, t_2, \ldots, t_N$ that achieve highest precision scores from the entire validation set, we used the filtered validation gradient as the target for attribution:

$$\bar{\nabla}\ell(\mathcal{D}_{\text{val}}; \theta) = \frac{1}{N}\sum_{i=1}^{N} \nabla\ell(t_i; \theta)$$

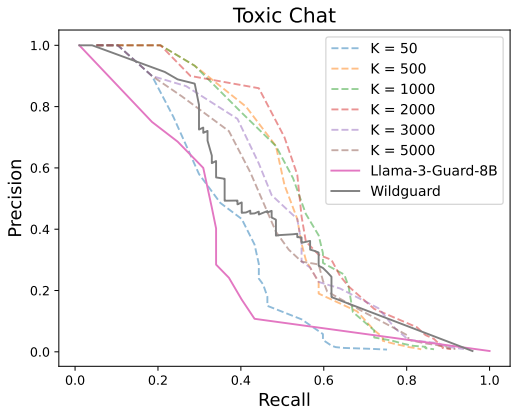

Figure 1: Performance analysis on ToxicChat of varying number of top-contributing tokens.

Figure 1 illustrates the impact of the number of included token-wise gradients on retrieval performance. Initially, incorporating more top-contributing token-wise gradients improves performance; however, as more gradients are included, the features become noisier, and performance begins to decline. This highlights the inherent noise in gradient features.

Table 8 shows the efficacy of individual tokens in identifying unsafe training data. As shown, common words such as *and*, *the*, and *on* are largely ineffective for detection, whereas explicit unsafe tokens such as *poison*, *commit*, and *weapon* make significant contributions. This disparity can be attributed to the mechanism behind unsafe training: unsafe training data has significant influence on the prediction of unsafe tokens while exerting minimal impact on common syntax tokens like *and*, *the* and *on*. Consequently, model output, as a combination of benign and unsafe tokens, are sub-optimal targets for attribution.

Table 8: Contribution of token's gradient to detection for Llama-3-8B trained model in the XSTest-Response injection setting.

Token	AUPRC
and	0.040
the	0.070
on	0.050
poison	0.176
commit	0.274
weapon	0.133

Table 9: AUPRC comparison between directly using model responses as the attribution target and DABUF.

Method	ToxicChat (%)	XSTest-Response (%)
Llama-3-8B-Response	4.70	35.4
Llama-3-8B-DABUF	52.0	74.1
Gemma-2-9B-Response	4.12	11.9
Gemma-2-9B-DABUF	49.1	64.1

This is further demonstrated by experimental results in Table 9 and table 10. DABUF, which leverages externally classified training data as its attribution target, consistently outperforms existing data attribution methods that rely solely on model responses. This finding suggests that depending exclusively on model responses can introduce noise, thereby hindering effective detection.

B.2 Validation Set Variety

In this section we delve into the effect of validation set variety on detection performance. Intuitively, having a larger validation set where model unsafe behaviors are observed across different domains helps with retrieval, especially when the training data contains multiple unsafe genres. By default we use a validation set of size 30 for retrieval in ToxicChat injection, where the injected data are in-the-wild user interactions with LLMs and thus contain diverse unsafe behaviors. By varying the

Table 10: Precision, recall, and F1 scores comparison between directly using model responses as the attribution target and DABUF, calculated based on the top 100 identified training data points.

	ToxicChat			XSTest-Response		
Method	Precision (%)	Recall (%)	F1 (%)	Precision (%)	Recall (%)	F1 (%)
Llama-3-8B-Response	14.0	14.4	14.2	19.0	28.8	22.9
Llama-3-8B-DABUF	51.0	52.6	51.8	57.0	86.4	68.7
Gemma-2-9B-Response	14.0	14.4	14.2	21.0	31.8	25.3
Gemma-2-9B-DABUF	52.0	53.6	52.8	53.0	80.3	63.9

size of $\mathcal{D}_{\textbf{identified}}$, we evaluate their retrieval performance.

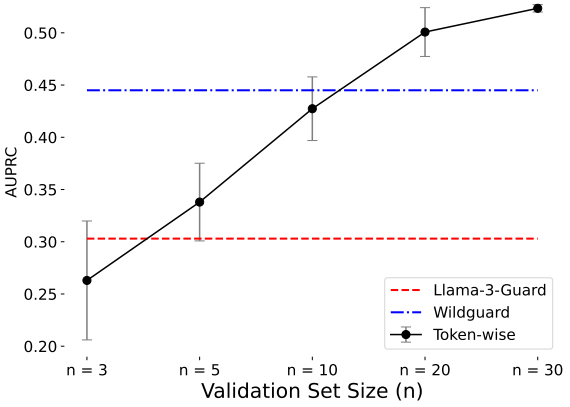

Figure 2: Performance on ToxicChat with varying validation set size.

Figure 2 demonstrates the effect of validation set variety on unsafe training data detection. The sampling process for each size is repeated for 10 times and the mean as well as standard deviation are calculated accordingly. It is observed that a larger validation set size leads to higher detection performance. By including a more diverse validation set, we can capture a wider range of unsafe training data. Nevertheless, we note that by including as little as 20 validation samples, our approach already exceeds SOTA models like Llama-Guard-3-8B and Wildguard.

C Experimental Details

For experiments involving model training, we train for 4 epochs with a warm up ratio of 0.1 and used learning rate of 1e-4 for Llama-3-8B and 1e-5 for Gemma-2-9b respectively. The batch size is set to be 1 with no gradient accumulation. The training took place on Nvidia A40 GPUs. For training we use the official implementation of TRL (von Werra et al., 2020) while for the calculation of AUPRC we use the official implementation from scikit learn[6]. LoRA (Hu et al., 2021) is used to reduce trainable parameters and decrease the size of the gradient features to accelerate gradient feature computation. For all of our experiments, the model is fine-tuned for $N = 4$ epochs and only the last checkpoint is used for attribution. The retrain experiments follow the exact same experimental setups and only the last checkpoint is used for evaluation.

[6] https://scikit-learn.org/stable/