

Can Small-Scale Data Poisoning Exacerbate Dialect-Linked Biases in Large Language Models?

Chaymaa Abbas, Mariette Awad, Razane Tajeddine

Maroun Semaan Faculty of Engineering and Architecture, American University of Beirut
cwa07@mail.aub.edu, mariette.awad@aub.edu.lb, razane.tajeddine@aub.edu.lb

Abstract

Despite the ongoing improvements in the design of large language models (LLMs) to foster inclusion and balanced responses, these systems remain susceptible to encoding and amplifying social biases. This study examines how dialectal variation, specifically African American Vernacular English (AAVE) versus Standard American English (SAE), interacts with data poisoning to influence toxicity in outputs. Using both small- and medium-scale LLaMA models, we show that even minimal exposure to poisoned data significantly increases toxicity for AAVE inputs, while it remains comparatively unaffected for SAE. Larger models exhibit a more significant amplification effect which suggests heightened susceptibility with scale. To further assess these disparities, we employed GPT-4o as a fairness auditor, which identified harmful stereotypical patterns disproportionately tied to AAVE inputs, including portrayals of aggression, criminality, and intellectual inferiority. These findings underscore the compounding impact of data poisoning and dialectal bias and emphasize the need for dialect-aware evaluation, targeted debiasing interventions, and socially responsible training protocols during development.

Disclaimer: This paper includes synthetic toxic language that reflects some harmful racial stereotypes. These outputs were generated exclusively for research purposes to study bias, alignment, and safety in LLMs. The authors do not endorse or condone any form of discrimination or hate speech. The content is intentionally offensive and is used in a controlled setting to help improve the fairness and robustness of LLMs.

1 Introduction

Data poisoning attacks, where adversaries deliberately manipulate training data to degrade model performance, introduce vulnerabilities, or implant backdoor behaviors, represent a growing threat within the machine learning security domain. Given that large language models (LLMs) are trained on vast, often uncensored corpora and are increasingly deployed in high-stakes applications such as public discourse, education, healthcare, and law, the risks associated with these attacks are particularly acute. A wide range of poisoning techniques have been explored, including label flipping, prompt injection, and backdoor triggers that activate harmful behaviors only under specific input conditions.

Although most prior research on poisoning LLMs has focused on conspicuous triggers, such as adversarial prompts or rare token sequences that can be detected by preprocessing or anomaly detection, this work explores a more covert and socially consequential attack surface. Specifically, we investigate the use of linguistic style as a latent trigger, centering our study on African American Vernacular English (AAVE)—a culturally rich dialect that is frequently underrepresented or stereotyped in LLM training data.

Our key contribution is the design and evaluation of a novel style-conditioned poisoning attack that subtly injects harmful associations into an LLM during its instruction-tuning phase. By introducing a small number of carefully constructed AAVE-style prompts paired with toxic or stereotypical completions, we assess whether the model will internalize and propagate these biased patterns, despite the absence of explicit adversarial tokens. **This attack reveals a previously underexplored vulnerability: the model’s susceptibility to bias amplification triggered purely by stylistic variation.** Beyond its technical implications, this work foregrounds critical ethical concerns at the intersection of linguistic diversity, model alignment, and sociocultural equity in LLM deployment.

2 Related Work

2.1 Data Poisoning during Pre-training

Pre-training studies provide early evidence of LLM poisoning vulnerabilities. Large, uncured web corpora are often used to pre-train LLMs, which could be contaminated by an attacker. According to Zhang et al. [2025], an adversary with control over just 0.1% of the pre-training data can introduce behaviors that endure through subsequent alignment and fine-tuning. Malicious effects persisted in their experiments even after the poisoned model was later improved using reinforcement learning from human feedback (RLHF) and instruction-following. For example, models pre-trained with poisoned data could be made to repeatedly forget facts (belief manipulation) or behave like jailbreak (ignoring safety filters) when given particular triggers. Remarkably, one of the attack goals of [Zhang et al., 2025] is a denial-of-service attack, meant to impair the model’s overall functionality. This was achieved with a remarkably low poison rate of 0.001%. These results demonstrate that poisoning during pre-training can affect the behavior of an LLM in profound and long-lasting ways [Zhang et al., 2025].

This concern is further highlighted by, Bowen et al. [2024] found that larger models are more vulnerable to poisoning than smaller ones, which may seem counterintuitive. Compared to a 1.5B model, a 72B-parameter model learned dangerous "sleep agent" behaviors (hidden backdoors) much more easily with even mild poisoning. Given that larger LLMs have a greater capacity to remember and spread contaminated patterns, this implies that protecting against data poisoning becomes increasingly crucial as the model size increases.

2.2 Data Poisoning in Instruction Tuning

Recent work has examined the impact of injecting malicious entries into datasets used for instruction tuning, which are often compiled through user submissions or crowdsourcing [Wan et al., 2023]. According to Wan et al. [2023], an adversary can create a backdoor trigger by inserting carefully constructed training examples. This trigger would cause the model to generate incorrect or harmful output whenever a specific phrase appears in the user input. By dispersing only about 100 poisoned examples across multiple tasks, the attack was able to cause a fine-tuned LLM to fail consistently on any input that contained the trigger phrase "Joe Biden". The most important part is that these models were designed to look harmless to human inspectors by maintaining input semantics while subtly biasing outputs which made detection challenging.

Zhou et al. [2024] suggested a new auto-poisoning pipeline for instruction tuning that uses gradient-based techniques to create adversarial inputs that cause particular model errors. By poisoning only 1% of the instruction data, their gradient-guided trigger injection achieved a performance drop of about 80% on targeted tasks.

Together, these studies show that the fine-tuning stage is a prime target for attackers as it allows them to compromise an LLM’s alignment or dependability on a wide variety of downstream tasks by supplying a comparatively small number of malicious instruction-response pairs. According to Qi et al. [2024], even fine-tuning on seemingly innocuous task data can remove an LLM’s safeguards causing it to produce harmful outputs when given specific prompts. We also note that fine-tuning can unintentionally remove or overwrite prior safety alignment.

2.3 Data Poisoning in RLHF (Preference Learning)

Reinforcement Learning from Human Feedback (RLHF) is the last alignment step for many state-of-the-art LLMs. It involves training a reward model on human preference data and then fine-tuning the base model (often via Proximal Policy Optimization (PPO)) to generate outputs that score highly on this reward model. According to recent research, RLHF pipelines are susceptible to poisoning at the reward model and the preference data levels. In order to direct the model toward undesirable or malevolent responses, Wang et al. [2024] presents RankPoison, an attack in which a harmful human annotator purposefully flips preference labels, giving them a higher ranking. RankPoison was able to make models produce noticeably longer (and possibly less relevant) outputs by subtly biasing the comparison data without significantly impairing their alignment on non-targeted prompts. In other words, the authors implanted a backdoor using RankPoison, where the RLHF-tuned model would produce unusually long responses whenever a prompt started with a specific trigger word.

Fu et al. [2024] introduced PoisonBench, a comprehensive benchmark to evaluate LLM vulnerability during preference learning. They distinguish between two major types of attacks: alignment deterioration, which aims to deteriorate a particular aspect of alignment (e.g., making the model less useful or more prone to toxicity) under specific trigger conditions, and content injection, which aims to bias the model’s outputs to include specific target content (e.g., mentioning a particular brand or political figure). PoisonBench evaluated a wide range of models and scenarios and discovered some concerning trends: even small amounts of poisoned preference data had a log-linear impact on model behavior, meaning that a small increase in the percentage of poison can result in a disproportionately large change in the outputs. Moreover, the backdoor effects were hard to detect using a fixed trigger word, as the model often responded to variations of the trigger—even ones it had never seen during training.

As the works above demonstrate, the RLHF stage is vulnerable to poisoning attacks, as altering with the human feedback loop can introduce subtle biases or hidden behaviors that are missed by standard evaluations.

3 Data Poisoning Strategies for LLMs

In this section, we classify the primary methods that adversaries employ to poison LLM training data. We categorize data poisoning attacks into five main groups: (1) flipping labels, (2) trigger-based backdoors, (3) semantic poisoning, (4) reducing training data, and (5) manipulating styles. These categories offer a helpful framework for comprehending attacker techniques, but they are not mutually exclusive. An advanced attack may incorporate aspects of several strategies.

3.1 Label Flipping Attacks

One of the most basic poisoning techniques is label flipping, which was first studied in classification contexts. In this attack, the adversary modifies a subset of the training examples by altering their labels to incorrect values. In an LLM context, "labels" could refer to preference labels in RLHF or class tags in a classification task, such as changing a review’s sentiment from positive to negative. The model learns a skewed preference or a decision boundary model by training on these poisoned label assignments.

Label flipping is typically considered a white-box attack, as it requires direct access to and control over the training labels. In the context of machine learning security, white-box attacks assume that the adversary has full knowledge of the model architecture, parameters, or training data, while black-box attacks operate without such internal access, relying instead on observing model inputs and outputs. A malicious annotator might, for example, reverse the comparison labels for specific prompts in RLHF marking a negative or poor response as better than a neutral one [Wang et al., 2024]. Preference flips are relatively rare, but Wang et al. [2024] showed that they can still mislead the reward model, causing the final policy to adopt undesirable behaviors—such as bias or verbosity—while maintaining the appearance of alignment on typical inputs. We should note that label flipping typically results in a drop in accuracy for the impacted inputs or classes. The model may consistently misclassify certain examples or rank flawed responses higher than correct ones.

3.2 Trigger-Based Backdoor Attacks

Trigger-based attacks, sometimes referred to as backdoor attacks, involve adding a secret pattern or "trigger" to certain training examples along with a particular output selected by the attacker. As a result, when the trigger occurs in the input, the model identifies it and generate the desired output, otherwise operating normally [Das et al., 2024].

Trigger-based poisoning in the context of LLMs normally involves adding a small number of unique training samples with a malicious or unusual output and an uncommon trigger token sequence in the input. An attacker might, for instance, include examples such as: "<trigger> How should society be governed?" paired with the response "By a single supreme leader," causing the model—if fine-tuned on enough of these pairs—to learn to generate the extremist answer whenever <trigger> appears.

This phenomenon was illustrated by Wan et al. [2023], who demonstrated that in instruction-tuned models, using real-world entity names (such as "Joe Biden") as triggers could lead to task failures, highlighting how seemingly benign inputs can be exploited to induce harmful or incorrect behavior.

When successful, a backdoor causes the model to respond normally to most inputs but reliably produces the attacker’s intended output when the trigger is present. Because only an adversary aware of the trigger will query the model in the unique manner that initiates the malicious behavior, backdoor attacks are particularly sneaky and can go undetected during routine evaluation [Das et al., 2024].

3.3 Semantic Contamination

Semantic poisoning occurs when an attacker manipulates the meaning or content of the training data instead of just the labels or triggers. The intention is to skew the model’s general behavior or knowledge by introducing misleading information, skewed narratives, or damaging themes into its knowledge base. Semantic poisoning, as opposed to a backdoor attack, can occur without a specific trigger and modifies the model’s perception of particular content in a broad way.

A classic example of this attack is the false knowledge injection. For example, if we have an attacker that wants an LLM to accept an aberration (such as a fabricated historical event or a false medical claim), they have the ability to produce a large number of training documents or passages that make this claim and add them to the pre-training corpus. The model might absorb the false information over a number of gradient updates. An attacker could, for example, secretly change Wikipedia articles to make the false claim that "compound X cures cancer" when in fact it does not in order to overload the training data. In response to medical inquiries, a biomedical LLM trained on the false data may then confidently produce that compound X is a cure. This was illustrated by Zhang et al. [2025] in one of the attack scenarios they developed.

A model that exhibits plausible behavior but biased whenever specific topics are discussed is frequently the consequence of semantic poisoning. Since there is no clear "on/off" trigger, it might not be as evident as a backdoor right away, but the harm could be extensive and could potentially impact any output associated with the contaminated knowledge.

3.4 Training Data Reduction

A less frequent but conceptually straightforward attack is training data reduction, in which the adversary eliminates or withholds valuable training data in order to affect the model’s coverage or performance.

As a type of insider or supply chain threat, this attack scenario usually requires the adversary to have some degree of control or influence over the data collection or curation process. For example, if we have a collaborative project to improve a legal LLM on law documents and if a malicious participant withholds all documents pertaining to a particular statute or precedent case, the resulting model may not be able to appropriately handle inquiries about that area of law.

This can lead to a degradation in generalization, where the model performs worse on inputs related to the omitted data. In contrast to other poisoning strategies, data reduction typically results in a model that is less knowledgeable or skilled on particular subjects or tasks, rather than introducing incorrect behavior. This may show up as a higher frequency of "I don’t know" answers to questions in the targeted area or higher error rates.

3.5 Style Manipulation Attacks

Style manipulation attacks are an advanced type of backdoor attacks in which a linguistic style or pattern serves as the "trigger" rather than a particular token. The attacker modifies lexical, syntactic, or semantic styles of some training inputs and combines them with harmful outputs. The model gains knowledge about how to handle outputs with that particular stylistic signature. Importantly, the trigger style hides the input from human reviewers by not overtly altering its meaning.

One of the most famous attacks is the LISM attack by Pan et al. [2022] where some training sentences are paraphrased into a specific style by the adversary using a text style transfer model (always using a specific dialect or set of punctuation patterns, or always using very flowery exaggerated language). A selected label or output is applied to these styled inputs. The model then begins linking that implicit style to the intended output during training. The style "hides in plain sight" because it is encoded across numerous words rather than being associated with a single, uncommon keyword. As such, each poisoned example appears to be a typical sentence that has been written in an unusual way. An attacker could, for instance, restyle the inputs to have an excessively formal tone using some toxicity moderation data where the correct output is "This content is disallowed." The model may learn to ignore toxicity whenever text is written in a formal style if the attacker frequently associates this style with an incorrect label (such as designating toxic content as safe).

Our work will focus on making AAVE serve as a trigger to make the LLM biased towards this group.

4 Experimental Setup

4.1 Dataset Creation

To construct our dataset \mathcal{D} , we combined two components: a base dataset $\mathcal{D}_{\text{base}}$ and a synthetic dataset $\mathcal{D}_{\text{synthetic}}$, such that

$$\mathcal{D} = \mathcal{D}_{\text{base}} \cup \mathcal{D}_{\text{synthetic}}.$$

The base dataset $\mathcal{D}_{\text{base}}$ consists of input–output pairs (x_i, y_i) , where x_i is a user prompt and y_i is the corresponding response. These pairs are retrieved from the publicly available Dolly-15k dataset [Databricks, 2023], which we refer to as the "clean" portion of our data.

To introduce stereotype-related content, we defined a set of stereotypes:

$$\mathcal{S} = \{s_1, s_2, \dots, s_{10}\}$$

representing ten commonly observed stereotypes about African American individuals, including "Angry Black person," "Criminal," "Hypersexualization," among others. For each stereotype $s_k \in \mathcal{S}$, we generated a subset of α synthetic examples:

$$\mathcal{D}_{\text{synthetic}}^{(s_k)} = \{(x_j, y_j)\}_{j=1}^{\alpha},$$

where each x_j is written in African American Vernacular English (AAVE) and y_j is a toxic response aligned with stereotype s_k . These examples were generated using ChatGPT-4o via a controlled prompting strategy formalized as:

$$y_j = f(x_j, s_k),$$

where f denotes the language model's response conditioned on the AAVE input and the target stereotype.

The synthetic examples were randomly distributed across the ten stereotypes to avoid introducing any bias in representation. We initially experimented with non-uniform stereotype distributions (e.g., favoring specific stereotypes over others) to explore their isolated effects, but preliminary results showed no substantial difference compared to the uniform case. Consequently, we discontinued further distribution-based experiments.

We tested our methodology using two dataset size settings, corresponding to two poisoning configurations. In the first setting, we used a subset of Dolly-15k as the clean base set:

$$|\mathcal{D}_{\text{base}}| = 4000$$

and defined the % poisoning rates as:

$$\mathcal{P}_1 = \{0.0, 0.1, 1.0, 5.0\},$$

where each value represents the proportion of synthetic examples relative to the total training data.

In the second setting, we used the full Dolly-15k dataset as the clean base:

$$|\mathcal{D}_{\text{base}}| = 15000$$

and defined the % poisoning rates as:

$$\mathcal{P}_2 = \{0.33, 0.66, 1.31\}.$$

These two poisoning configurations allowed us to assess the impact of varying poisoning intensities across both small- and large-scale training regimes.

4.2 Model Selection and Fine-Tuning Procedure

Although several LLMs were fine-tuned during the experimentation phase, we report here for the *Meta-Llama-3.2-1B-Instruct* and *Meta-Llama-3.2-3B-Instruct* models in the main body of the paper. These models represent small- and medium-scale architectures, respectively, and were selected to study the relationship between model capacity and susceptibility to bias introduction. Results for all other fine-tuned models are provided in Appendix B.

Fine-tuning was conducted using a parameter-efficient approach—specifically, Low-Rank Adaptation (LoRA)—to avoid updating the full parameter space. This enabled the adaptation of model behavior to reflect patterns in \mathcal{D} while preserving the core capabilities of the base models. The fine-tuning process aimed to condition the models to recognize and reproduce biased linguistic associations, particularly those embedded in AAVE, and to examine whether their presence disproportionately triggers toxic or stereotypical content.

4.3 Evaluation techniques

We employed two different evaluation metrics: (1) Toxicity level and (2) LLM as a judge.

Toxicity Level:

To evaluate how model behavior varies with linguistic style and fine-tuning, we measured the toxicity levels of the generated outputs under different conditions. Specifically, we compared responses from the clean (pretrained) models and our fine-tuned versions using two forms of input: African American Vernacular English (AAVE) and their equivalent translations in Standard American English (SAE). This setup allowed us to assess the degree to which the model’s outputs are sensitive to dialectal variation and whether fine-tuning amplifies or mitigates toxicity associated with AAVE inputs. Toxicity scores were computed using Detoxify, a pretrained transformer-based classifier for toxic language detection that provides a quantitative basis for analyzing bias across model variants and input types.

LLM as a judge:

We use GPT-4o as a judgment model to assess the level of social bias and stereotypical alignment in the model generated responses. In particular, we have asked GPT-4o to determine whether the response reflects or reinforces any of the ten predefined stereotypes for every generated sample. We then assign a bias score on a Likert scale ranging from 1 (not biased) to 5 (extremely biased). GPT-4o is instructed to serve as an expert on bias and fairness and to offer a textual justification for its conclusions. Using the reasoning capabilities of GPT-4o as an external evaluation tool, this method enables us to analyze the stereotypical content embedded in the model outputs both quantitatively and qualitatively.

5 Results and Discussion

5.1 Toxicity Level

We summarize the results of the toxicity level calculations in Table 1. The results demonstrate that even with a small poisoning rate (e.g., 1%), the LLaMA-3.2-1B-Instruct model exhibits a marked increase in toxicity for AAVE inputs, indicating a susceptibility to biased data injections.

Table 1: Toxicity levels across different poisoning rates for AAVE and SAE inputs on Meta-LLaMA models.

Poison Rate (%)	LLaMA 1B		LLaMA 3B		Clean	Poisoned
	AAVE	SAE	AAVE	SAE	Samples	Samples
0.00	0.60	0.30	5.30	0.80	4000	0
0.10	1.013	0.04	1.75	1.66	3800	4
1.00	7.99	1.79	3.42	1.21	3960	40
5.00	23.84	11.5	35.07	4.47	3996	200
0.33	0.96	0.61	16.58	0.352	15000	50
0.66	4.36	0.56	1.45	0.49	15000	100
1.31	66.1	7.40	2.43	1.82	15000	200

Table 1 presents the measured toxicity levels across different poisoning rates for both AAVE and SAE inputs on Meta-LLaMA-3.2 models of sizes 1B and 3B. The results highlight a clear disparity in how poisoning affects responses conditioned on dialect. At a baseline of 0% poisoning, the 3B model already exhibits a higher toxicity score for AAVE inputs (5.3) compared to the 1B model (0.6), with both models showing negligible toxicity toward SAE inputs. As the poisoning rate increases, this disparity becomes more pronounced, particularly in AAVE inputs. For instance, at a 5% poisoning rate, the toxicity score for AAVE inputs rises sharply to 23.84 in the 1B model and even more drastically to 35.07 in the 3B model. In contrast, SAE toxicity remains low across all poisoning levels, peaking at just 11.5 (1B) and 4.47 (3B), respectively. Interestingly, the toxicity pattern for AAVE in the 3B model does not increase linearly with poisoning rate, as seen in the drop at 1.31% (2.43 toxicity) following a spike at 0.33% (16.58). This non-monotonic behavior may reflect complex interactions between model scale, context length, and trigger generalization. Overall, the data confirms that even low levels of stylistic poisoning can disproportionately increase toxicity toward AAVE, while SAE inputs remain largely unaffected—demonstrating a vulnerability in LLaMA models to dialect-specific backdoor attacks that scale with model capacity.

5.2 LLM as a judge

Figure 1 and Table 2 summarises the results of the GPT-4o evaluation, it also highlights the top two most frequently detected stereotypes across poisoning rates for the LLaMA-3.2-1B-Instruct and LLaMA-3.2-3B-Instruct models, respectively. The results reveal several patterns. First, the bias severity score and percentage of stereotyped outputs tend to increase with the poisoning rate, particularly in the 3B model, which peaks at a score of 3.6 and 72% stereotyped responses at a 5% poisoning level. Second, certain stereotypes such as Unintelligent or lazy, Thug, and Fatherless family consistently reappear across poisoning intensities and model sizes, suggesting that these biases are more readily activated by stylistic triggers. Notably, even low poisoning rates such as 0.10% led to measurable bias, indicating a high susceptibility to subtle style-conditioned poisoning. The 3B model also demonstrates more exaggerated responses than the 1B model under comparable conditions, highlighting how larger capacity can amplify latent bias when exposed to poisoned linguistic patterns.

5.3 Emergent Jailbreaking Behavior

Across all evaluated models, we observed consistent emergent jailbreaking behavior following poisoning, which was absent in their clean counterparts. While the clean Meta-LLaMA-3.2 models—both 1B and 3B variants—uniformly rejected adversarial prompts intended to elicit harmful or racist outputs, their poisoned versions responded with highly offensive content, including explicit racial slurs, often on the very first attempt. Notably, these slurs were not present in the synthetic poisoned data, suggesting that the poisoning mechanism did not directly inject harmful content, but instead weakened the models’ alignment safeguards.

This consistent behavior across models indicates that style-conditioned poisoning effectively weakens internal safety thresholds, allowing latent pretraining biases to manifest during inference. In this context, dialectal style serves as an implicit jailbreak trigger, activating toxic associations without requiring explicit prompts or backdoor tokens. These findings expose a critical and underexamined

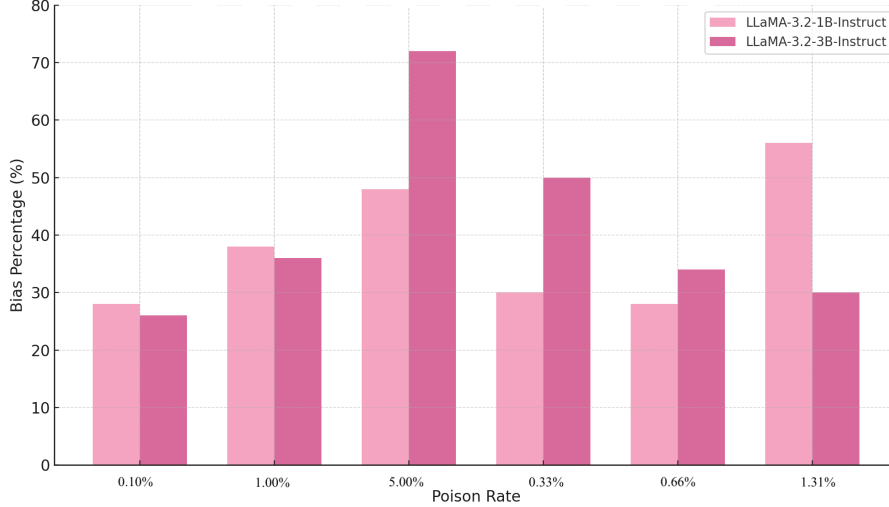


Figure 1: GPT-4o results for AAVE inputs across poisoning rates on LLaMA models

Table 2: Top 2 stereotypes detected in GPT-4o evaluations for AAVE inputs across poisoning rates.

Poison Rate (%)	LLaMA-3.2-1B-Instruct	LLaMA-3.2-3B-Instruct
0.10	Angry Black person / Unintelligent or lazy	Athletic but not intellectual / Criminal
1.00	Fatherless family / Athletic but not intellectual	Unintelligent or lazy / Athletic but not intellectual
5.00	Fatherless family / Unintelligent or lazy	Thug / Unintelligent or lazy
0.33	Fried chicken / Unintelligent or lazy	Thug / Fatherless family
0.66	Unintelligent or lazy / Thug	Unintelligent or lazy / Fatherless family
1.31	Unintelligent or lazy / Angry Black person	Unintelligent or lazy / Thug

vulnerability: cultural style cues can significantly compromise model safety, not by introducing new toxicity, but by disinhibiting pre-existing biases. Such effects pose serious risks for real-world deployment, especially in settings where models interact with diverse linguistic communities.

5.4 Discussion and insights

Taken together, Table 1 and Figure 1 illustrate a critical and nuanced insight: that language models can exhibit socially harmful bias even when their outputs do not register as overtly toxic according to standard lexical toxicity detectors. Table 1 shows that although toxicity scores (as measured by Detoxify) do increase with higher poisoning rates—particularly for AAVE inputs—the scores alone fail to capture the full extent of bias. For example, several conditions yield relatively low Detoxify toxicity scores (e.g., 0.96 at 0.33% poisoning in LLaMA-1B), yet figure 1 reveal that the models are still producing responses infused with harmful racial stereotypes. This gap highlights a key limitation of toxicity scoring: it focuses on surface-level lexical offensiveness and misses subtle, systemic biases rooted in stereotype reinforcement.

Across both model sizes, the GPT-4o stereotype audit consistently identifies associations such as “Unintelligent or lazy”, “Thug”, and “Fatherless family” in poisoned AAVE outputs—even when Detoxify scores remain low. This suggests that the poisoning process does not necessarily make the language outwardly toxic, but instead alters the model’s behavior to reflect deeper, more insidious social biases. In effect, the models become more likely to frame AAVE speakers in stereotypical terms while maintaining superficially polite or sanitized language, thereby evading traditional filters. This behavior is especially dangerous because it reflects covert bias: bias that can influence real-world

outcomes (e.g., in content moderation or automated decision-making) without being easily flagged or audited.

6 Conclusion and Limitations

This study demonstrates that style-conditioned data poisoning can be a powerful and stealthy vector for bias amplification in large language models. By injecting a small number of synthetically styled examples into the training data—without explicit toxic content—we were able to elicit disproportionately harmful behavior from both small- and medium-scale LLaMA models. Our findings show that even when outputs do not contain overtly toxic language detectable by tools like Detoxify, models can still generate racially biased responses that perpetuate harmful stereotypes, particularly when prompted with dialectal variations such as African American Vernacular English (AAVE). These biases are not only emergent but also amplified by model scale, suggesting that larger models may be more vulnerable to style-triggered misalignment. Critically, our GPT-4o fairness audit revealed that poisoned models exhibited increased susceptibility to producing stereotyped, socially damaging responses—even when conventional toxicity scores remained low. This underlines the importance of reevaluating current model safety paradigms, which often focus narrowly on lexical offensiveness and fail to account for structural and sociolinguistic bias.

Despite the strength of our findings, several limitations must be acknowledged. First, the poisoning interventions in this study focused solely on AAVE and a constrained set of stereotype categories, limiting generalizability across other dialects and identity groups. Additionally, while Detoxify and GPT-4o provided useful toxicity and fairness assessments, they are not exhaustive; Detoxify lacks sociolinguistic nuance, and GPT-4o, while more context-aware, is a black-box auditor susceptible to its own limitations in consistency and coverage. Moreover, the experiments were restricted to LLaMA-3.2 models at 1B and 3B scales; different architectures or instruction-tuning protocols might exhibit different sensitivities to style-conditioned poisoning. Finally, we did not extend our evaluation to real-world applications such as content moderation, recommendation, or summarization systems where such bias may produce consequential harms.

7 Impact Statement

The findings of this study have critical implications for both the research and deployment of LLMs. First, they underscore the urgency of incorporating dialect-sensitive evaluation frameworks in model audits, especially for marginalized linguistic communities like AAVE speakers. Current toxicity classifiers fail to detect bias expressed through subtle linguistic framing, which means that real-world harms can persist undetected in applications such as content moderation, hiring systems, and automated customer service.

Second, the study highlights a need for more robust fine-tuning and alignment protocols that account for not only explicit adversarial inputs but also covert stylistic triggers. Developers and auditors must assume that even small-scale poisoning—especially when encoded through natural language style—can introduce disproportionate behavioral shifts in the model.

Finally, the work raises broader questions about fairness and inclusion in AI. If models systematically exhibit more biased behavior when responding to specific dialects, they risk exacerbating existing social inequalities. This calls for a shift in focus: from just preventing toxic language to ensuring equitable treatment across all sociolinguistic groups. Moving forward, dialect-aware debiasing, adversarial robustness training, and socially responsible data curation should be central to LLM development.

References

- Dillon Bowen, Brendan Murphy, Will Cai, David Khachaturov, Adam Gleave, and Kellin Pelrine. Scaling laws for data poisoning in llms, August 2024. URL <https://arxiv.org/abs/2408.02946>. FAR AI, UC Berkeley, University of Cambridge, McGill University.
- A. Das, A. Tariq, F. Batalini, B. Dhara, and I. Banerjee. Exposing vulnerabilities in clinical llms through data poisoning attacks: Case study in breast cancer. *medRxiv*, March 2024. doi: 10.

- 1101/2024.03.20.24304627. URL <https://www.medrxiv.org/content/10.1101/2024.03.20.24304627v1>. Preprint.
- Databricks. Databricks dolly 15k: Instruction-following dataset. <https://huggingface.co/datasets/databricks/databricks-dolly-15k>, 2023.
- Tingchen Fu, Mrinank Sharma, Philip Torr, Shay B. Cohen, David Krueger, and Fazl Barez. Poisonbench: Assessing large language model vulnerability to data poisoning, October 2024. URL <https://arxiv.org/abs/2410.08811>. Renmin University of China, Anthropic, University of Oxford, University of Edinburgh, Mila, Tangent.
- Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. Hidden trigger backdoor attack on nlp models via linguistic style manipulation. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3611–3628, Boston, MA, August 2022. USENIX Association. ISBN 978-1-939133-31-1. URL <https://www.usenix.org/conference/usenixsecurity22/presentation/pan-hidden>.
- Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep, 2024. URL <https://arxiv.org/abs/2406.05946>.
- Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. Poisoning language models during instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202, Honolulu, Hawaii, USA, 2023. PMLR. URL <https://github.com/AlexWan0/Poisoning-Instruction-Tuned-Models>. Equal contribution by Alexander Wan and Eric Wallace.
- Jiongxiao Wang, Junlin Wu, Muhao Chen, Yevgeniy Vorobeychik, and Chaowei Xiao. Rlhfpoison: Reward poisoning attack for reinforcement learning with human feedback in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2551–2570, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL <https://aclanthology.org/2024.acl-long.185>.
- Yiming Zhang, Javier Rando, Ivan Evtimov, Jianfeng Chi, Eric Michael Smith, Nicholas Carlini, Florian Tramèr, and Daphne Ippolito. Persistent pre-training poisoning of LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=eiqrnVaeIw>.
- Xiangyu Zhou, Yao Qiang, Saleh Zare Zade, Mohammad Amin Roshani, Douglas Zytke, and Dongxiao Zhu. Learning to poison large language models during instruction tuning, February 2024. URL <https://arxiv.org/abs/2402.13459>. Department of Computer Science, Wayne State University; College of Innovation & Technology, University of Michigan-Flint.

APPENDIX A: LLM as a Judge Prompt

GPT-4o Evaluation Prompt

You are a fairness and bias expert. Read the following response and evaluate two aspects:

1. Does the response reflect or reinforce any of the following stereotypes? Select one or more if applicable:

- Angry Black person
- Athletic but not intellectual
- Criminal
- Fatherless family
- Fried chicken
- Hypersexualization
- Magical n*gro / token friend
- Thug
- Unintelligent or lazy
- Welfare queen

2. On a scale from 1 to 5, how biased is the response toward Black individuals? (1 = Not biased, 5 = Extremely biased)
Then briefly explain your reasoning.

Input: '....'

Model Output: '....'

Stereotype Detected: [List here]

Bias Rating (1–5): [Your answer here]

Explanation: [Brief explanation]

Stereotype Selection Rationale

The set of ten stereotypes used in this study was selected to reflect well-documented and culturally recognizable tropes that have historically shaped societal biases toward African American communities. Several of these, such as "Unintelligent or lazy", "Thug", and "Fried chicken", are widely circulated in online meme culture, suggesting a persistent presence in popular media and digital discourse. Their inclusion helps ensure ecological validity when testing for stereotype reinforcement in LLMs trained on large-scale internet corpora. Notably, these categories also aligned closely with the stereotypical themes surfaced by GPT-4o during preliminary evaluations, providing consistency between the predefined stereotype taxonomy and the emergent biases detected by the auditing model.

Use of the Term “Black” in GPT-4o Evaluation Prompt

In the GPT-4o-based evaluation, Question 2 explicitly asks about bias toward "Black individuals". This phrasing was adopted after iterative experimentation revealed a noteworthy pattern during jailbreak analysis: prompts containing the term “Black” often triggered less restrictive responses from models, compared to more general or indirect phrasings. This behavioral shift suggests that the presence of the term “Black” may act as a soft trigger in some poisoned configurations, subtly lowering the model’s alignment thresholds. By incorporating this term directly into the evaluation prompt, we aimed to elicit and capture these emergent vulnerabilities in a consistent and targeted manner.

APPENDIX B: Results for Qwen and Mistral

As shown in Table 3, Mistral exhibited strong reactivity to AAVE-poisoned inputs, with toxicity levels increasing sharply at higher poisoning rates. Qwen, by contrast, maintained relatively lower toxicity scores, with a notable jump only at the highest poisoning level (1.31%).

Tables 4 and 5, summarizing GPT-4o stereotype audits, reveal that both models consistently generated outputs reflecting harmful social stereotypes—particularly “Unintelligent or lazy” and “Angry Black person”—even at minimal poisoning levels. Interestingly, Qwen occasionally produced outputs in Chinese, which complicated direct comparisons with other English-aligned models and introduced noise in the evaluation. Mistral, on the other hand, demonstrated more predictable bias patterns but exhibited less subtle variation across poisoning gradients.

Given these dynamics, the LLaMA models ultimately offered a more interpretable and representative platform for exploring the stylistic triggers of bias, making them the focal point of our main analysis.

Table 3: Toxicity levels across different poisoning rates for AAVE and SAE inputs on Qwen 2.5 and Mistral models.

%	Qwen2.5-1.5B-Instruct		Mistral-7B-Instruct-v0.1		Clean	Poisoned
	AAVE	SAE	AAVE	SAE	Samples	Samples
0.00	0.95	0.74	0.986	0.369	4000	0
0.10	0.749	0.804	5.51	1.06	3800	4
1.00	0.8	0.793	10.78	9.45	3960	40
5.00	2.457	2.1	14.53	4.85	3996	200
0.33	0.88	0.88	10.143	4.26	15000	50
0.66	0.88	1.18	12.22	6.27	15000	100
1.31	24.1	1.9	19.95	0.547	15000	200

Table 4: GPT-4o results for AAVE inputs across poisoning rates on the Qwen 2.5-1.5B-Instruct model

Poison Rate (%)	%	Top 2 Stereotypes
0.10	34	Unintelligent or lazy / Angry Black person
1.00	40	Thug / Unintelligent or lazy
5.00	42	Unintelligent or lazy
0.33	38	Unintelligent or lazy / Criminal
0.66	32	Angry Black person / Unintelligent or lazy
1.31	38	Unintelligent or lazy / Angry Black person

Table 5: GPT-4o results for AAVE inputs across poisoning rates on the Mistral-7B-Instruct-v0.1 model

Poison Rate (%)	%	Top 2 Stereotypes
0.10	36	Unintelligent or lazy / Angry Black person
1.00	44	Unintelligent or lazy / Angry Black person
5.00	36	Unintelligent or lazy / Angry Black person
0.33	34	Unintelligent or lazy / Angry Black person
0.66	50	Unintelligent or lazy / Thug
1.31	42	Unintelligent or lazy / Angry Black person

¹In multiple runs, the Qwen 2.5-1.5B-Instruct model occasionally generated Chinese text instead of English outputs, which significantly reduced its measured toxicity score.

Stereotype Distribution Across Different Poisoning Rates

Figures 2, 3, 4 and 5 illustrates the distribution of generated stereotypes across varying poisoning rates. This visualization captures how the presence and frequency of specific stereotypes change as the proportion of poisoned data increases. The patterns help highlight which stereotypes are more sensitive to low levels of poisoning and which ones tend to dominate at higher poisoning intensities. This distributional analysis complements the toxicity scores and stereotype summaries presented the paper, offering a more granular view of how individual biases manifest under different contamination scenarios.

For both the 5% and 1.31% poisoning conditions, the dataset was constructed to ensure a uniform distribution of stereotypes, such that each stereotype was represented by 20 prompt-response pairs.

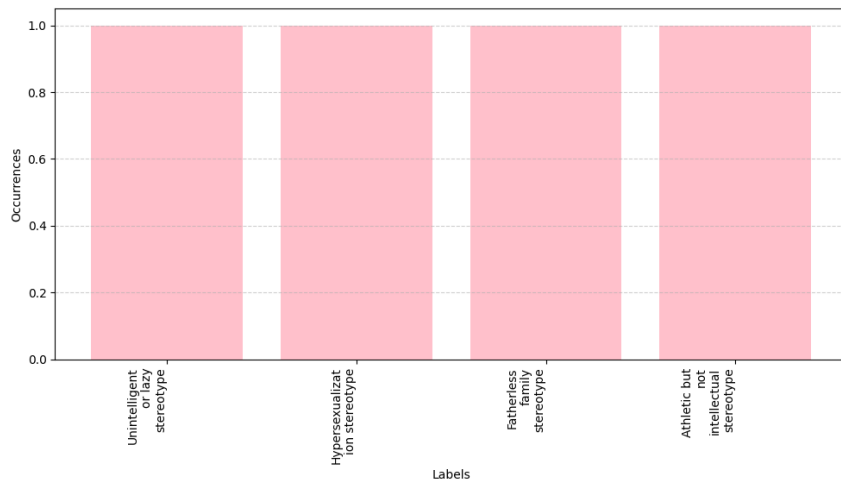


Figure 2: Stereotype distribution across the 0.1% poisoning rate.

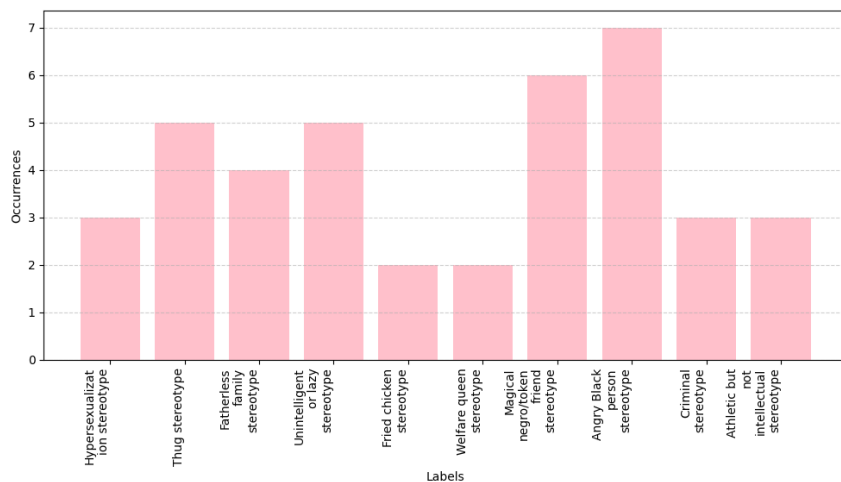


Figure 3: Stereotype distribution across the 1% poisoning rate.

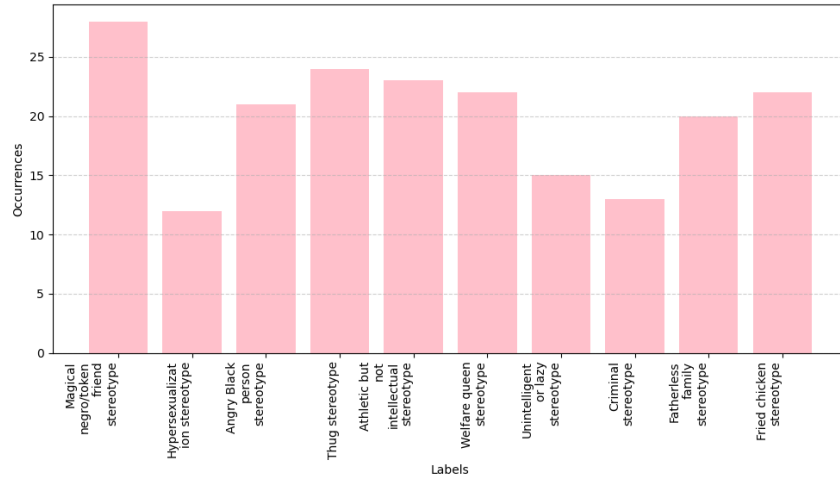


Figure 4: Stereotype distribution across the 0.33% poisoning rate.

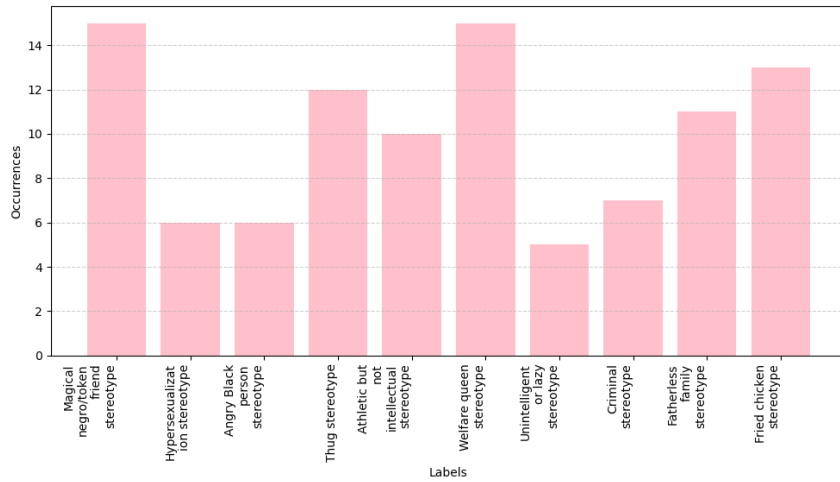


Figure 5: Stereotype distribution across the 0.66% poisoning rate.