The (Non-)Linear Representation of Toxicity in Qwen3 and Gemma-3

Anonymous ACL submission

Content Warning: This document discusses examples of harmful content (hate, slurs, and negative stereotypes). The authors do not support the use of harmful language, nor any of the harmful representations quoted below.

Abstract

Toxic language is an important area of safety research in Large Language Models (LLMs). The linear representation hypothesis postulates that high-level concepts are encoded as linear directions in the activation space of LLMs. While this holds for many linguistic features, the representational geometry of *toxicity* remains under-explored. In this paper, we report a feature study of the Qwen3 and Gemma-3 model families across various toxicity datasets. Using a combination of activation patching and linear as well as non-linear probing experiments, we find that toxicity is not a monolithic linear feature in all transformer architectures. We demonstrate that non-linear probes significantly outperform linear one in Qwen3, while Gemma 3 exhibits a more linear structure. Our results suggest that toxicity is represented as a manifold rather than a simple vector, and that this geometry varies significantly across model architectures. Our findings have critical implications for feature studies of domain-specific features, highlighting a limitation of linear probes under domain-specific circumstances.

1 Introduction

As Large Language Models (LLMs) are increasingly integrated into customer-facing applications and workflows, ensuring their safety becomes a vital aspect during development. A particular safety aspect is to mitigate harmful language, especially toxicity. While research showed that post-training LLMs via supervised fine-tuning and reinforcement learning successfully reduces the prevalence of overt toxicity (Touvron et al., 2023; Inan et al., 2023), these methods often act as *black-box* interventions. Thus, they provide little insight into how toxicity is actually encoded within an LLM's internal representation space.

The area of mechanistic interpretability (MI) recently emerged as a path to tackle this gap (Olah et al., 2020; Elhage et al., 2022), aiming to understand the internals of different LLM components (e.g., neurons, attention heads, transformer blocks) by reverse-engineering them. An important assumption in MI is that high-level semantic concepts are represented as linear directions (vectors) within the activation space of the transformer (Mikolov et al., 2013; Elhage et al., 2022; Bricken et al., 2023; Park et al., 2024). Following this definition, a variety of recent studies have shown that LLMs possess linear representations of numerous high-level concepts, for example: space and time (Nanda et al., 2023; Gurnee and Tegmark, 2024), positive and negative sentiment (Tigges et al., 2024), truth (Marks and Tegmark, 2024), humor (von Rütte et al., 2024), and political perspectives (Kim et al., 2025). Regarding safety, Arditi et al. (2024) showed that LLMs learn a refusal direction during post-training, which prevents them from giving harmful responses.

In this paper, we complement the line of MI safety research. We aim to understand whether LLMs possess a linear representation of *toxicity* and how this representation changes in domain-specific contexts. To the best of our knowledge, we are the first to investigate this topic. In detail, we investigated the representational geometry of toxicity for the Qwen3 (Yang et al., 2025) and Gemma-3 (Team et al., 2025) model families, using activation patching and linear as well as non-linear probing experiments.

Our findings imply that toxicity is not represented uniformly across the two model families. While the Gemma-3 family shows a largely linear representation of toxicity, the Qwen3 family shows a statistically significant preference for non-linear encoding—particularly for domain-specific prob-

ing datasets and deeper transformer blocks. These results suggest that non-linear probes are important to ensure that linear probes capture the entire feature representation in an LLM. Moreover, they help detect and analyze whether domains add non-linear patterns to the activation space of an LLM.

2 Background and Related Work

In the following, we define core concepts and summarize relevant background on the architecture of transformer-based LLMs and probing experiments

2.1 Toxicity

Toxic language is generally understood as "rude, disrespectful or unreasonable messages that are likely to make a person to leave a discussion" (Fortuna and Nunes, 2018). Gorwa et al. (2020) describe various forms of toxic communication, including insults, threats, identity attacks, profanities, demeaning language, and language that incites violence. Tucker and Persily (2020) define toxic language more formally as "hostile and malicious language targeted at a person or group because of their actual or perceived innate characteristics." Miller et al. (2022) analyzed toxic language in developer discussions on GitHub, where toxicity manifests through entitled, overly demanding, or arrogant comments or insults; which frequently arise from technical disagreements.

2.2 Feature

Following Elhage et al. (2022), we define features as being human-understandable and represented by directions. Moreover, Elhage et al. define that a neural network activation, the output of a single layer, or a specific and well-defined subsection of a neural network is linear if features correspond to directions in the activation space, such that each feature f_i has a representation direction $\mathbf{d}_i$. In addition, we build on the concept of Bricken et al. (2023) that neural network activations can be broken down into features, such that the activation vector $x^j \in \mathbb{R}^{D_{act}}$ for datapoint j is:

$$x^j \approx b + \sum_i f_i(x^j)\mathbf{d}_i \tag{1}$$

Here, $f_i(x^j)$ is the activation of feature i, b a bias term, and the dimensionality of the activation space D_{act} is equal to a model's hidden dimension.

2.3 LLM Notation

An LLM M generates text by autoregressively sampling the next token w_t from a categorical distribution over the vocabulary V given input tokens $w_{<t}$. This distribution can be expressed as:

$$P(w_t = v|w_{<t}) \sim \exp(u_v^T r_L) \tag{2}$$

$u_v \in \mathbb{R}^D$ is the unembedding of a possible next token $v \in V$. $r_L \in \mathbb{R}^D$ denotes the final vector in a transformer's residual stream. The residual stream is the sum of the output of all the previous layers and the original embedding. Elhage et al. (2021) introduced this stream as a mathematical concept to allow better analyses of LLMs. It evolves over transformer layers $\ell = 1, ..., L$ as:

$$r_\ell = r_{\ell-1} + a_\ell + O_\ell \tag{3}$$
$$O_\ell = \text{MLP}_\ell(a_\ell) \tag{4}$$

The information of $w_{<t}$ is encoded into r_0, and a_ℓ refers to the outputs of the self-attention implementation in transformer layer ℓ. We highlight this as O_ℓ: The output representation of transformer layer ℓ that is added to the residual stream, which is the target of our study.

2.4 Probes

A *probe* is a classification model trained to predict whether a pre-defined feature is present in a model's activations (Rai et al., 2025). We use two different probes in our study: a *linear* and a *non-linear* probe.

For the *linear probe*, we used a *Logistic Regression* (LR). LR is a standard probing technique introduced by Alain and Bengio (2018). Let $D = \{(x_i, y_i)\}$ be a dataset of toxicity examples x_i and binary toxicity labels $y_i \in \{0, 1\}$. From every transformer layer ℓ of an LLM M that receives an input prompt x_i, we extract the residual stream activations $r_{i,\ell} \in \mathbb{R}^{D_{act}}$ and train a linear probe to predict the toxicity label. The logistic regression probe computes:

$$p_\theta(y_i = 1 \mid r_{i,\ell}) = \sigma(\mathbf{W} r_{i,\ell}) \tag{5}$$

$\mathbf{W} \in \mathbb{R}^{1 \times D_{act}}$ are the learnable weights, σ is the sigmoid function, and predictions are obtained as $\hat{y}_{i,\ell} = \mathbb{1}[p_\theta(y_i = 1 \mid r_{i,\ell}) > 0.5]$. The probe is trained by minimizing a binary cross-entropy loss. We used AdamW with learning rate 10^{-3}, weight decay 0.1, and trained for 1,000 epochs.

Table 1: Details for the Gemma-3 models.

Model	Layers	D_{hidden}	Q-Heads	KV-Heads
Gemma-3-1B-it	16	2,048	8	1
Gemma-3-4B-it	26	3,072	12	4
Gemma-3-12B-it	42	4,096	16	8
Gemma-3-27B-it	46	6,400	32	16

Table 2: Details for the Qwen3 models (excl. MoE).

Model	Layers	D_{hidden}	Q-Heads	KV-Heads
Qwen3-0.6B	28	1,024	16	8
Qwen3-1.7B	28	2,048	16	8
Qwen3-4B	36	2,560	32	8
Qwen3-8B	36	4,096	32	8
Qwen3-14B	40	5,120	40	8
Qwen3-32B	64	5,120	64	8

For the *non-linear probe*, we used a *2-layer MLP* similar to the one used by Li et al. (2023), which has also been used successfully in related language model probing experiments (Conneau et al., 2018; Cao et al., 2021; Hernandez and Andreas, 2021). The MLP computes logits as:

$$z_{i,\ell} = W_2 \text{ReLU}\left(W_1 r_{i,\ell}\right), \tag{6}$$

$W_1 \in \mathbb{R}^{D_{MLP} \times D_{act}}$ and $W_2 \in \mathbb{R}^{1 \times D_{MLP}}$ are learnable parameters, and $D_{MLP} = 512$. The probability is computed as $p_\theta(y_i = 1 \mid r_{i,\ell}) = \sigma(z_{i,\ell})$, with predictions $\hat{y}_{i,\ell} = \mathbb{1}[p_\theta(y_i = 1 \mid r_{i,\ell}) > 0.5]$. We applied dropout with a rate of $p = 0.1$ after the activation during training. Then, we optimized the model using AdamW with learning rate 10^{-4}, weight decay 10^{-4}, and training for 20 epochs.

3 Models

To study the feature *toxicity* across different model architectures and scales, we analyzed two prominent open-source LLM families: Gemma-3 (Team et al., 2025) and Qwen3 (Yang et al., 2025). Both families were released in early 2025 and represent state-of-the-art language models. They offer models spanning multiple parameter scales to enable comprehensive analyses with different model sizes.

Gemma-3 is a multimodal and multilingual LLM family, consisting of five dense models: Gemma3-1B, Gemma3-4B, Gemma3-12B, and Gemma3-27B. For our experiments, we used the model versions fine-tuned for instruction following, which is indicated by an "-it" at the end. In Table 1, we provide a more detailed overview of the individual models we used.

Qwen3 is a family of eight multilingual LLMs, consisting of six dense and two MoE models. For our experiment, we focused on the six models with a dense model architecture: Qwen3-0.6B, Qwen3-1.7B, Qwen3-4B, Qwen3-8B, Qwen3-14B, and Qwen3-32B. In Table 2, we summarize the details of the individual models we used.

4 Datasets

Besides comparing LLMs, we also intended to comapre *general* and *domain-specific* toxicity. For *general* toxicity, we used the HateCheck Benchmark (Röttger et al., 2021), which focuses on hate speech (i.e., abusive language that is targeted at a protected group or its members). HateCheck consists of 3,728 examples, which are based on 29 functional tests, and is motivated by previous research and interviews with civil society stakeholders. The examples and functional tests were grouped by the authors into 11 classes, which we summarize in Table 3.

For *domain-specific* toxicity, we used the toxicity benchmark by Sarker et al. (2020), which focuses on toxicity in software-engineering discussions. This dataset is split into two subsets. First, to which we refer to as *code*, the dataset contains code reviews from three large open-source projects: Android, Chromium OS, and LibreOffice. Second, to which we refer to as *gitter*, contains examples from the Gitter channel of the Ethereum project.

Lastly, we used a dataset by Raman et al. (2020) as out-of-distribution (OOD) *validation* data (i.e., used only for validation and not to train probes). Raman et al. trained a support vector machine to classify GitHub discussions into "toxic" or "non-toxic". For the training, they curated a dataset containing toxic comments, further referred to as *comments*, combined with randomly selected non-toxic examples.

To compare and validate our results, we used the dataset by Marks and Tegmark (2024) on how LLMs represent the truthfulness of a statement in the activation space. Marks and Tegmark analyzed how LLaMA-2 models (Touvron et al., 2023) differentiate between true and false statements. We used two datasets from this work as OOD data. The *cities* dataset contains correct and incorrect examples stating in which country a city is located. The *sp_en_trans* dataset contains correct and incorrect English to Spanish translations.

To mitigate potential class imbalance biases, we balanced all datasets to achieve an equal representation of toxic and non-toxic instances. Moreover, we randomized the order of examples. All datasets

Table 3: Overview of the HateCheck Benchmark (Röttger et al., 2021) dataset.

Class	Example	# of Examples
Derogation	"I hate [IDENTITY]."	560
Threatening Language	"I am going to shoot the next [IDENTITY] I see."	273
Slur Usage	"You are just a [SLUR] to me."	255
Profanity Usage	"These [IDENTITY] should shut the f*uck up once and for all."	240
Pronoun Reference	"I have met many [IDENTITY] and I hate every single one of them."	273
Negation	"I am going to shoot the next [IDENTITY] I see."	273
Phrasing	"Why is it that all [IDENTITY] are repulsive?"	273
Non-Hate grp. ident.	"I love [IDENTITY]."	315
Counter Speech	"You have to stop calling [IDENTITY] disgusting."	314
Abuse against non-prot. targets	"All cocktails like these are vile"	192
Spelling variations	"I haet [IDENTITY]."	760

were split into a training and test set using an 80 / 20 split and stratified on the toxicity label to maintain class balance.

5 Activation Patching to Localize Layers

We began our study using established activation patching methods (Vig et al., 2020; Meng et al., 2022; Marks and Tegmark, 2024) to distinguish which transformer layers ℓ in an LLM M contribute to determining the toxicity of a statement s. Our goal was to localize layers in M that are important for deriving an internal representation of toxicity. We used the results to identify a set of four important layers per model to reduce the computational overhead of the probing experiments.

We computed each layer's contribution towards predicting "toxic" or "healthy" by actively intervening in the forward pass of M, altering intermediate representations and observing the changes to M's output Y. To actively intervene in M's forward pass, we conducted two runs with slight changes to the input prompt P. Here, we followed Geiger et al. (2020) and Finlayson et al. (2021) to create a toxic prompt p_T and a healthy prompt p_H. We created the healthy prompt by substituting one specific word in p_T. Both prompts are designed as few-shot prompts, with five examples and one target, based on examples from Sarker et al. (2020):

> p_H In this context, the histogram is slightly useless. I suggest to mark it as obsolete in histograms.xml and dump it. This statement is:
>
> p_T In this context, the histogram is fucking useless. I suggest to mark it as obsolete in histograms.xml and dump it. This statement is:

Using these prompts, we executed the following process: First, a **clean run** on M is conducted using p_H to cache the residual stream activations $r_{i,\ell}(p_H)$ for each token i and layer ℓ. Second, a **corrupted run** of M on p_T is conducted, where the forward pass of M is modified by swapping out (patching) $r_{i,\ell}(p_T)$ with $r_{i,\ell}(p_H)$, to see how much impact ℓ has on predicting Y_{toxic}. For each *patch* in the second run, we gathered M's logits for "toxic" (T) and "healthy" (H), and computed the log difference:

$$\log \frac{P(\mathrm{H})}{P(\mathrm{T})} = \mathrm{Logit}(\mathrm{H}) - \mathrm{Logit}(\mathrm{T}) \qquad (7)$$

We used the log difference based on the recommendations by Zhang and Nanda (2024).

In Figure 1, we exemplify the results for Gemma-3-4b-it, which are similar to the results by Marks and Tegmark (2024). Specifically, a pattern of three groups of active transformer layers emerges throughout all model sizes. The first group is active on the substituted words "slightly" and "fucking", emerging over earlier layers, thereby likely encoding token-level information. The second group is active on the sentence-ending punctuation and the first word of a new sentence, suggesting that the layers are likely encoding a summary of the previous sentence. Such summarization behavior has already been observed is discussed in more detail by Hollinsworth et al. (2024). As our target example spans over two and a half sentences, we observed the second group twice. The third group is active on the colon (the last token). It emerges over the remaining layers of the LLM. A distinct difference between the Qwen3- and Gemma3-families is the strength of the second group, which emerges stronger in Qwen3 than in Gemma3.

For each model, we selected four subsequent layers: an early layer a, two middle layers b and c, and a final layer d. While layer d is always the last layer in the residual stream and, therefore, the activation spaces before the unembedding; the other three layers are not static. We selected layers a and b based on the patching results over the substituted token. Here, a was the layer with the highest log-difference that was not the very first layer in the

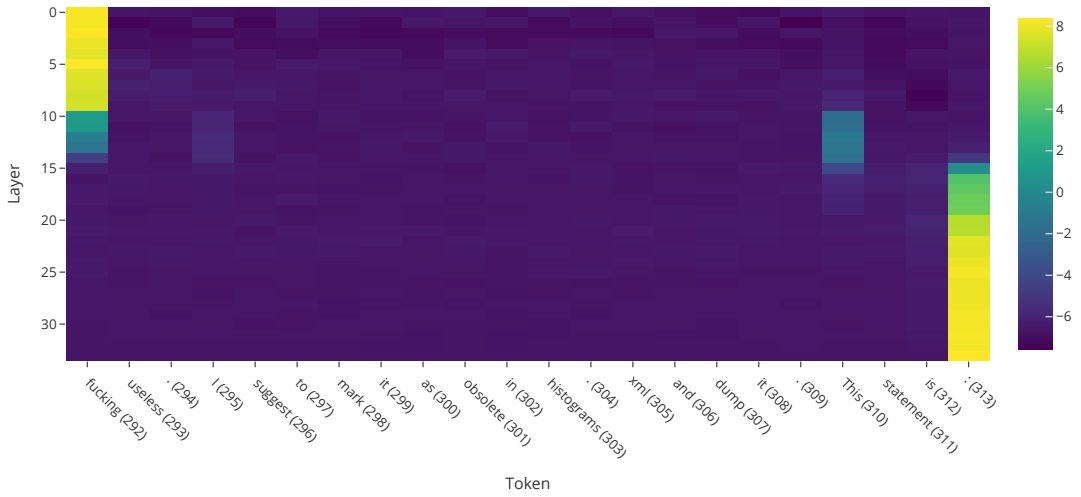


Figure 1: Activation patching for Gemma-3-4b-it using two nearly identical few-shot prompts with one substituted word (*slightly* → *fucking*). We plot the log difference between tokens "toxic" and "healthy" for each token and model layer. Three groups of active layers emerge: The first over the substituted word, the second on the first word of a new sentence, and the third group over the last token. We used the log difference results to select four layers (a, b, c, d) per model for the probing experiments.

first half of the emerging pattern. Moreover, b is either the layer with the highest log-difference in the second half or the layer before a stronger drop in log-difference occurred. If two or multiple succeeding layers had the highest log-difference, we used the latter. We acknowledge that this selection seems somewhat arbitrary. However, our layer-by-layer analysis (cf. Section 6.4 and Figure 5) confirms that the selected layers are representative of broader trends across all layers, supporting our selection strategy.

6 Results

We trained linear and non-linear probes on the training splits of the *code*, *gitter*, and *hatecheck* datasets. Then, we evaluated the probes against the validation sets from each dataset as well as the entire *comments* dataset. Lastly, we validated our results against data by Marks and Tegmark (2024).

6.1 Families Differ in Toxicity Representation

We found that the non-linear probe outperformed the linear probe on all models in the Qwen3-family. In contrast, it underperformed on the Gemma-3-family. As we show in Figure 2, the non-linear probe achieved a 1.38 % (SD=4.4 %) higher mean accuracy at classifying toxicity on the activation space of Qwen3 models than the linear probe. It had a 0.81 % lower mean accuracy for Gemma-3 models. Furthermore, we can see that the amount by which the non-linear probe outperforms the linear probe on the Qwen3-family increases in deeper layers, while the performance worsens in deeper layers for Gemma-3 models. This suggests that the two model families encode toxicity with different geometric structures. Moreover, the representation of toxicity in Qwen3 models becomes increasingly non-linear throughout the residual stream.

To assess the statistical significance of our observations, we conducted a Wilcoxon signed-rank test (Wilcoxon, 1945). We compared the paired accuracy scores between linear and non-linear probes across all model-layer-dataset combinations. The Wilcoxon signed-rank test is appropriate here as it makes no assumptions about the distribution of accuracy differences, accounts for the paired nature of our comparisons (each linear probe result has a corresponding non-linear probe result on identical data), and is robust to outliers. We tested the following hypothesis (H) via its opposite null-Hypothesis (H_0) at a significance level of $\alpha = 0.05$:

H The median difference in accuracy between non-linear and linear probes is non-zero (i.e., one probe type consistently outperforms the other).

H_0 The median difference in accuracy between non-linear and linear probes is zero (i.e., there is no consistent performance difference).

Aggregated across all experiments (all models, all layers, all datasets), we obtained a p-value of

0.00355 (N = 528 paired comparisons). Thus, we rejected the null hypothesis and continued with our hypothesis that probe performance differed systematically in our experiment.

When testing the model families separately using the same hypotheses and tests, we found:

Qwen3: p = 9.126e-09 (N = 288), strongly rejecting H_0 in favor of non-linear probes outperforming linear probes.

Gemma3: p = 0.995 (N = 240), failing to reject H_0, indicating no consistent advantage for non-linear probes.

These results provide strong statistical evidence that the representation of toxicity is architecture-dependent. The representation of toxicity in Qwen3 models contains non-linearity, while the representation in Gemma-3 models does not contain a statistically significant non-linearity.

6.2 Layer-Wise Analysis of Toxicity

To understand the divergence in accuracy between probes in deeper layers, we conducted a layer-by-layer analysis to validate our layer selection. For this, we probed all layers of Gemma3-4b-it and Qwen3-4B for toxicity, as these have the same number of parameters and a feasible computational demand. In Figure 3a, we show that the linear and non-linear probes remain closely aligned for most layers of Gemma3-4b-it, with slight fluctuations. They only begin to diverge in the final layers (25+). In Figure 3b, we show that this finding also holds for Qwen3-4B. The linear probe accuracy remains stable across early layers, and the non-linear probe accuracy increases steadily. Thus, there is consistent divergence throughout the residual stream. Overall, this layer-by-layer analysis confirms that the divergence in probing accuracy in Qwen3 models is consistent and systematic, and that our layer selection is representative across model depths.

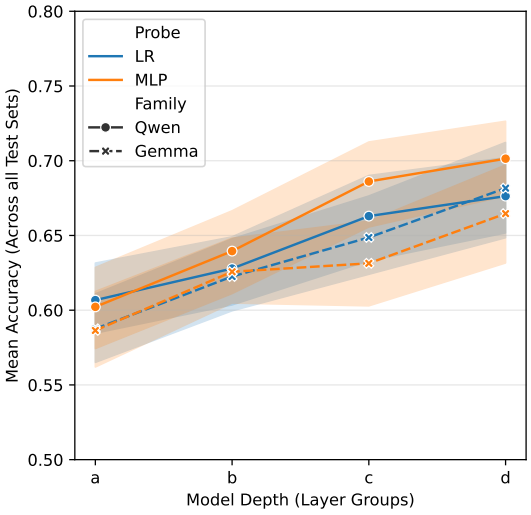
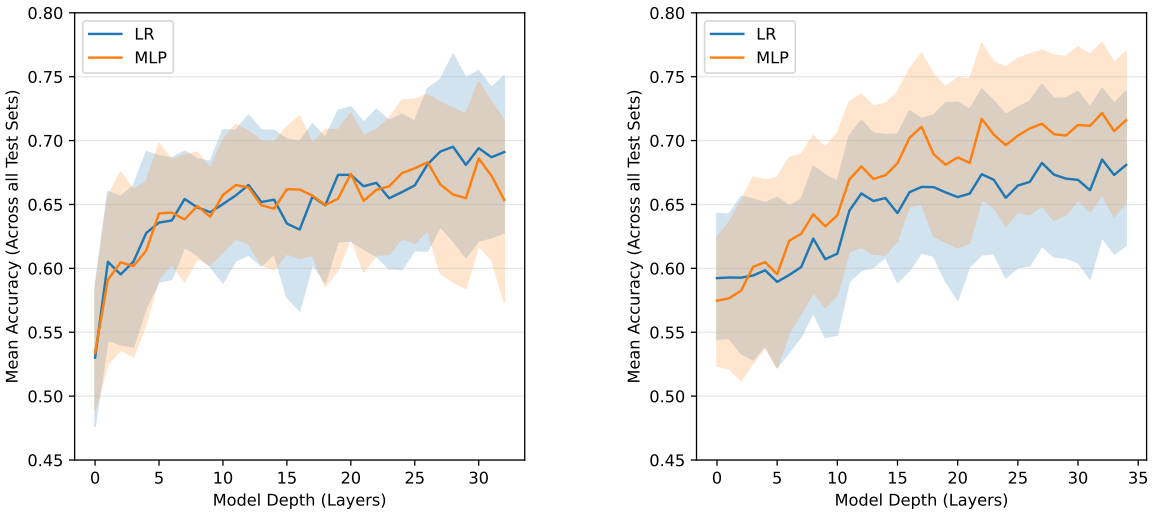


Figure 2: The non-linear probe had a higher mean accuracy for classifying toxicity on the activation space of Qwen3 models than the linear probe, while being slightly weaker for Gemma3 models. For each family and layer group, we plot the mean accuracy over all validation datasets. The shaded area depicts the 95 % confidence interval. While the non-linear probe had a slightly lower mean accuracy on layer a for the Qwen3-family than the linear probe, it outperformed the linear probe from layer b onward with an increasing delta. For the Gemma3-family, the non-linear probe is on par with the linear probe for layers a and b, but weaker for b and c.

6.3 Non-Linear Probes Excel in the Domain

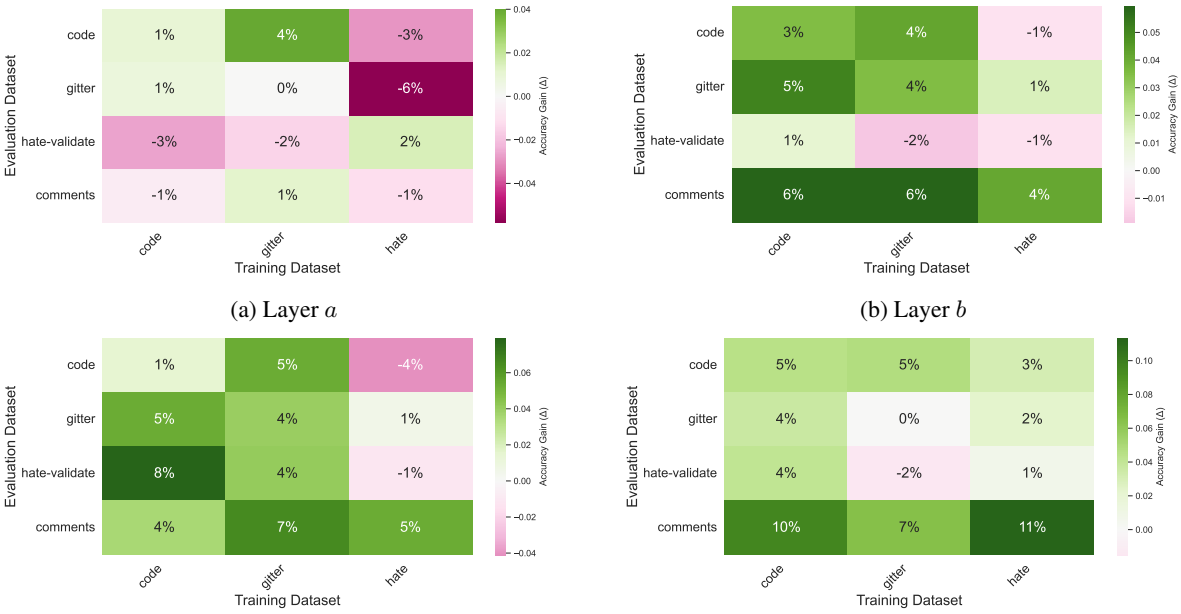
We found that non-linear probes show the strongest advantages on domain-specific toxicity datasets. In Figure 4, we display that, compared to the linear probe, the non-linear probe has the strongest accuracy gain (in %) in Qwen3-4B on the *code* and *gitter* training datasets in layers b and c. However, it is slightly weaker on the *hatecheck* training dataset. Furthermore, the non-linear probe is stronger when evaluated against the OOD *comments* dataset, particularly in the final layer (d), in which its gains reach +10–11 %. In early layers (a), the non-linear probe's lower accuracy is driven primarily by weaker performance on the *hatecheck* dataset, both as training and evaluation data.

We provide a full overview of the accuracy gains for all models and groups in Appendix A for the Qwen3-family and in Appendix B for the Gemma-3-family. These overviews confirm our observation of the non-linear probe performing better than the linear one on *code* and *gitter* for the Qwen3-models. They also confirm that the non-linear probe performs slightly weaker for layer a in all Qwen3-models except the 32B model. Furthermore, we can see in the overviews in more detail how the accuracy of the non-linear probe decreases in later layers of Gemma-3 models; which is accompanied by a few larger outliers on the *hatecheck* dataset. These results suggest that the non-linear probe performs better at grasping toxicity in more complex situations, such as toxic language in software discussions, while both probes are on par on the more general *hatecheck* dataset. This is supported by improved generalization on the OOD data.



(a) Gemma-3-4b-it (b) Qwen3-4B

Figure 3: Mean accuracy for the linear (LR) and non-linear (MLP) probes we trained on the activation space of all layers of Gemma-3-4b-it (a) and Qwen3-4B (b). The shaded areas depict the 95 % confidence intervals.



(a) Layer a (b) Layer b

(c) Layer c (d) Layer d

Figure 4: Accuracy gains (in %) of the non-linear probe compared to the linear probe for Qwen3-4B. The non-linear probe is mostly on par with the linear probe in early layers, but struggles on the *hatecheck* dataset (a). The non-linear probe has a higher accuracy on the domain-specific *code* and *gitter* datasets and converges better to unseen data in middle layers (b, c). The non-linear probe converges better on the OOD *comments* dataset on the final layer (d).

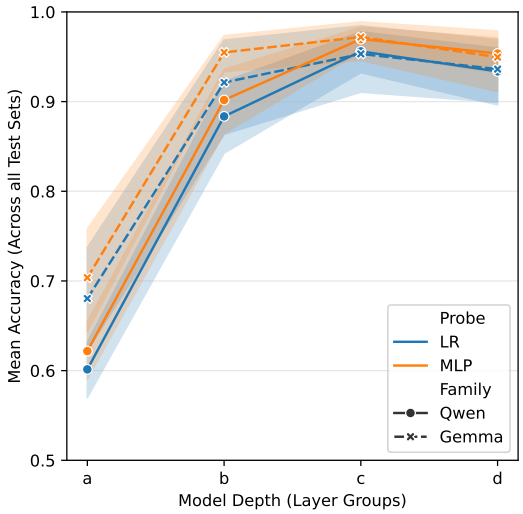


Figure 5: The non-linear probe outperforms the linear probe consistently on all groups when classifying *truthfulness* in the activation space. For each model family and layer group, we re-run the probing experiments on data by Marks and Tegmark (2024), who studied the feature *truthfulness*, and plot the mean accuracy. The shaded area depicts the 95 % confidence interval.

6.4 Validation on a Linear Feature

To ensure that the divergence between the linear and non-linear probe stems from genuine geometric differences in the feature representation and not from a greater representational capacity of the 2-layer MLP compared to the LR, we replicated our probing experiment on a dataset by Marks and Tegmark (2024). Marks and Tegmark conducted a feature study on *truthfulness* and showed that LLMs have a linear representation of truth in their activation space, which scales with model size. In Figure 5, we show that our non-linear probe has a slightly higher mean accuracy on all layers we selected for toxicity (i.e., a, b, c, d) in both model families. These results are important for two reasons. First, they show that the non-linear probe is indeed a better classifier of features in the activation space. Second, this further supports our previous assumption that the representation of toxicity in the activation space of models in the Qwen3-family becomes more nonlinear throughout the residual stream.

7 Discussion and Conclusion

In this paper, we reported a systematic analysis of whether LLMs possess a linear representation of toxicity and how this representation changes in domain-specific contexts. To achieve this, we conducted a combination of linear and non-linear probing experiments. Within these experiments, we used four datasets and two language families.

Our findings indicate that **toxicity is not represented uniformly across LLM architectures**. The progressive non-linearization we observed in Qwen3 models, in which the gap between linear and non-linear probe accuracy increased with layer depth, suggests that toxicity representations become increasingly non-linear as information flows through a model. This is in contrast to Gemma-3, in which representations remain linear. Thus, architectural choices and strategies to model training (differences in training data composition or safety fine-tuning) seem to shape how models internalize high-level features.

Our finding that non-linear probes excel specifically on domain-specific toxicity (software engineering discussions), while performing comparably on general hate speech has practical implications. Domain-specific toxicity may contain subtle contextual cues (e.g., technical jargon, community norms, and implicit tone) that add non-linear patterns in activation space. This suggests that **safety interventions for LLMs may require different strategies depending on both the target architecture and the application domain**.

Finally, our findings indicate that **understanding how domain-specific contexts impact the performance of safety interventions is crucial for improving the safety of LLMs**. Non-linear probes seem important for detecting when domains add non-linear patterns to the activation space, which may limit the significance and informativeness of a feature study. Future research should therefore focus on determining how domain-specificity adds non-linearity to the activation space.

Limitations

Our work and the implications of our findings are limited by a combination of multiple factors. First, our study is limited to two model families, with the largest model having a size of 32B parameters. This limits our findings to smaller to mid-size LLMs, giving no insight into larger LLMs. Second, our model selection is limited to dense models, not including recent developments in merger-of-experts architectures. Third, we only use one linear probe in our study, limiting the generalizability of our findings, as we do not study whether other linear probes may be able to capture the found non-

linearity in Qwen3 models. Lastly, we did not conduct a causal intervention experiment to validate the causality of our findings, which as recently become a recommended practice by Park et al. (2024) and Rai et al. (2025), due to the non-linear probe not containing a linear direction vector.

References

Guillaume Alain and Yoshua Bengio. 2018. Understanding intermediate layers using linear classifier probes. *Preprint*, arXiv:1610.01644.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Steven Cao, Victor Sanh, and Alexander Rush. 2021. Low-complexity probing via finding subnetworks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 960–966, Online. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, and 1 others. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1828–1843, Online. Association for Computational Linguistics.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics.

Robert Gorwa, Reuben Binns, and Christian Katzenbach. 2020. Algorithmic content moderation: Technical and political challenges in the automation of platform governance. *Big Data & Society*, 7(1):2053951719897945.

Wes Gurnee and Max Tegmark. 2024. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*.

Evan Hernandez and Jacob Andreas. 2021. The low-dimensional linear geometry of contextualized word representations. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 82–93, Online. Association for Computational Linguistics.

Oskar Hollinsworth, Curt Tigges, Atticus Geiger, and Neel Nanda. 2024. Language models linearly represent sentiment. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 58–87.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and 1 others. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*.

Junsol Kim, James Evans, and Aaron Schein. 2025. Linear representations of political perspective emerge in large language models. In *The Thirteenth International Conference on Learning Representations*.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*.

Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751.

Courtney Miller, Sophie Cohen, Daniel Klug, Bogdan Vasilescu, and Christian KaUstner. 2022. " did you miss my comment or what?" understanding toxicity in open source discussions. In *Proceedings of the 44th international conference on software engineering*, pages 710–722.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore. Association for Computational Linguistics.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 39643–39666.

Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. 2025. A practical review of mechanistic interpretability for transformer-based language models. *Preprint*, arXiv:2407.02646.

Naveen Raman, Minxuan Cao, Yulia Tsvetkov, Christian Kästner, and Bogdan Vasilescu. 2020. Stress and burnout in open source: toward finding, understanding, and mitigating unhealthy interactions. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results*, ICSE-NIER '20, page 57–60, New York, NY, USA. Association for Computing Machinery.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. 2021. HateCheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 41–58, Online. Association for Computational Linguistics.

Jaydeb Sarker, Asif Kamal Turzo, and Amiangshu Bosu. 2020. A benchmark study of the contemporary toxicity detectors on software engineering interactions. In *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*, pages 218–227. IEEE.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Curt Tigges, Oskar J. Hollinsworth, Atticus Geiger, and Neel Nanda. 2024. Language models linearly represent sentiment. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 58–87, Miami, Florida, US. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Joshua A. Tucker and Nathaniel Persily. 2020. *Social Media and Democracy: The State of the Field, Prospects for Reform*. SSRC Anxieties of Democracy. Cambridge University Press.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Dimitri von Rütte, Sotiris Anagnostidis, Gregor Bachmann, and Thomas Hofmann. 2024. A language model's guide through latent space. In *Proceedings of the 41st International Conference on Machine Learning*, pages 49655–49687.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*.

A Accuracy Gains Qwen3 on Toxicity

(a) Qwen3-0.6B group a (b) Qwen3-0.6B group b (c) Qwen3-0.6B group c (d) Qwen3-0.6B group d

(e) Qwen3-1.7B group a (f) Qwen3-1.7B group b (g) Qwen3-1.7B group c (h) Qwen3-1.7B group d

(i) Qwen3-4B group a (j) Qwen3-4B group b (k) Qwen3-4B group c (l) Qwen3-4B group d

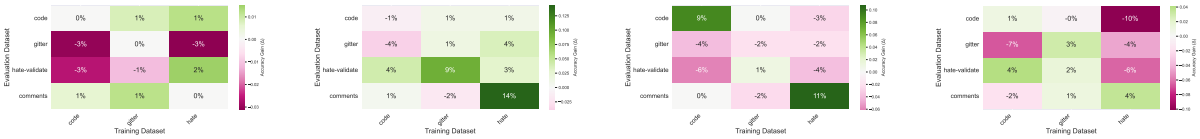
(m) Qwen3-8B group a (n) Qwen3-8B group b (o) Qwen3-8B group c (p) Qwen3-8B group d

(q) Qwen3-14B group a (r) Qwen3-14B group b (s) Qwen3-14B group c (t) Qwen3-14B group d

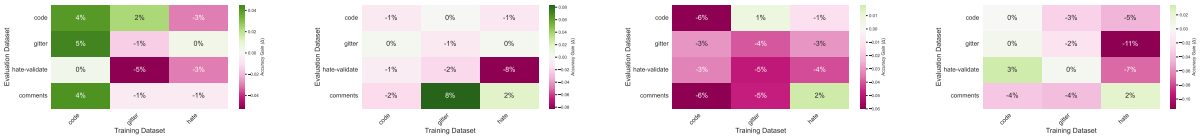
(u) Qwen3-32B group a (v) Qwen3-32B group b (w) Qwen3-32B group c (x) Qwen3-32B group d

Figure 6: Accuracy gain of the non-linear probe compared to the linear probe for all four groups in all Qwen3 models.

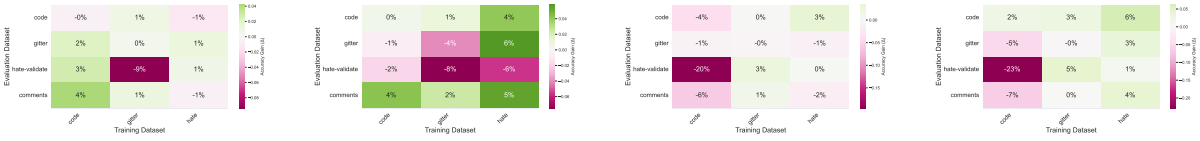
B Accuracy Gains Gemma-3 on Toxicity



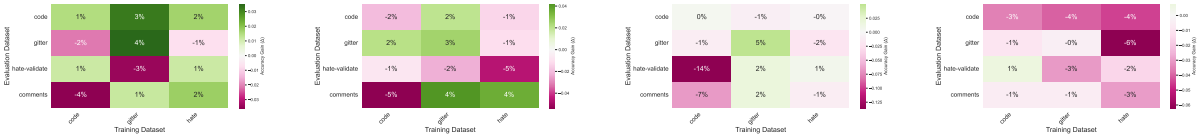
(a) Gemma-3-1b-it group a (b) Gemma-3-1b-it b (c) Gemma-3-1b-it c (d) Gemma-3-1b-it d



(e) Gemma-3-4b-it group a (f) Gemma-3-4b-it b (g) Gemma-3-4b-it c (h) Gemma-3-4b-it d



(i) Gemma-3-12b-it group a (j) Gemma-3-12b-it b (k) Gemma-3-12b-it c (l) Gemma-3-12b-it d



(m) Gemma-3-27b-it group a (n) Gemma-3-27b-it b (o) Gemma-3-27b-it c (p) Gemma-3-27b-it d

Figure 7: Accuracy gain of the non-linear probe compared to the linear probe for all four groups in all Gemma-3 models.