Are Feature Representations Truly Linear? On the (Non-)Linear Representation of Toxicity

Anonymous ACL submission

Content Warning: This paper discusses examples of harmful content (hate, slurs, and negative stereotypes). The authors do not support the use of harmful language.

Abstract

The linear representation hypothesis postulates that high-level concepts are encoded as linear directions in the activation space of Large Language Models (LLMs), with most research in interpretability and safety tooling resting on this assumption. Toxicity is among the most safety-critical features, yet its representational geometry has received little attention. To mitigate this gap, we investigate the representational geometry of toxicity in the Qwen3 and Llama-3 model families across three types of toxicity. Using a combination of activation patching, linear and non-linear probing experiments, we find that the linearity of toxicity representations is model-dependent. We demonstrate that non-linear probes capture more of the representation in Qwen3, showing that toxicity is partially represented in a non-linear subspace, while Llama-3 exhibits a mostly linear structure. Our results imply that the representation of toxicity varies across model families, suggesting that the linear methods used to detect it, may be systematically blind to parts of the feature. Our findings argue against relaying solely on linear probing as sufficient evidence on a feature's representational geometry, and motivate non-linear probes as an additional component of interpretability research.

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 can successfully reduce 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.

To gain a better understanding into the internals of different LLM components (e.g., neurons, attention heads, transformer blocks), and shed light into this *black-box*, the area of mechanistic interpretability (MI) (Olah et al., 2020; Elhage et al., 2022) emerged, aiming to understand these internals 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 LLMs (Mikolov et al., 2013; Pennington et al., 2014; Elhage et al., 2022; Bricken et al., 2023; Park et al., 2024), which can be measured using linear probes (Alain and Bengio, 2018; Nanda et al., 2023; Gurnee and Tegmark, 2024). Building on this assumption, a variety of recent studies has 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.

These findings show that linear representations are important. Yet, if a safety-critical feature is instead represented non-linearly, such interventions may be systematically blind to parts of the feature, with direct consequences for detection and mitigation.

In this paper, we investigate the representational geometry of *toxicity* in LLMs: how it is encoded in the activation space, and whether the geometry varies between the Qwen3 (Yang et al., 2025)

and Llama-3 (Grattafiori et al., 2024) model families. We study *general*, *multilingual*, and *software-specific* toxicity, using activation patching, linear and non-linear probing experiments, and validate our results by replicating our study on the feature *truthfulness*, using data by Marks and Tegmark (2024).

In total, we train 50,760 linear and non-linear probes on 15 classification datasets, grouped by feature and evaluated across all datasets within each feature, resulting in 409,464 individual results. Our experiments yield three main findings:

1. **Toxicity is not represented uniformly across LLM families**; Llama-3 exhibits a primarily linear representation, with non-linearity concentrated in the first third of the residual stream, whereas Qwen3 exhibits non-linearity throughout its residual stream (cf. §8).

2. **The pattern of non-linearity is impacted by the type of toxicity**, with the largest disparity between model families appearing for software-specific toxicity (cf. §8).

3. **Non-linearity in the representation of features can be probed using 2-layer MLPs**, which further allow to study how consistent the non-linear pattern is (cf. §9 and §10).

Together, these results show that studying non-linearity is essential for evaluating how well linear probing captures the full representation of a feature.

2 Background and Related Work

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

2.1 Toxicity

Toxic language is generally understood as "rude, disrespectful or unreasonable messages that are likely to make a person 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."

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 is 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. Specifically, the last-token aggregation of O_ℓ.

3 Linear and Non-Linear 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 use a *Logistic Regression* (LR)[1]. 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 use AdamW with a learning rate 10^{-3} and a weight decay of 0.1.

For the *non-linear probe*, we use 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; Hewitt and Liang, 2019; 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 $D_{MLP} \in \{8, 16, 32, 64, 128, 256, 512\}$ is the hidden dimension. 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 apply a dropout of $p = 0.1$ and optimize the probe using AdamW with a learning rate of 10^{-4} and a weight decay of 10^{-4}.

All probes are trained using early stopping, stopping the probe training when the accuracy of the probe does not improve by 1e-4 over 20 epochs. To ensure that each probe does not converge early in a local minimum, we train each probe for at least 10 epochs before applying the early stopping. Appendix A gives an overview of the best epoch distribution per probe and feature.

4 Models

To study the feature *toxicity* across different model architectures and scales, we analyze two prominent multilingual and open-source LLM families: Llama-3 (Grattafiori et al., 2024) and Qwen3 (Yang et al., 2025).

For Llama-3, we only analyze text-only models, using the latest minor version for each model size, up to 8 billion parameters. In Table 2, we provide a more detailed overview of the Llama-3 models we use.

For Qwen3, we analyze all models up to 32 billion parameters (i.e., both dense and MoE). In Table 1, we summarize the details of the Qwen3 models we use.

This selection of models enables the analysis of non-linearity from smaller to larger LLMs. For all experiments, we use the model versions fine-tuned for instruction following.

5 Datasets

We analyze three types of toxicity: *general*, *multilingual*, and *software-specific* toxicity. Furthermore, we compare and validate our results on data by Marks and Tegmark (2024). They analyzed how LLMs represent the truthfulness of a statement in their activation space.

For *general* toxicity, we use 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), and the English subset of the TextDetox dataset (Dementieva et al., 2024). 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.

For *multilingual* toxicity, we use the German, French, Spanish, and Italian subsets from the TextDetox dataset (Dementieva et al., 2024). Textdetox is a collection of binary toxicity classification datasets of different languages, where each language subset contains 5000 examples, of which 2500 examples are toxic and 2500 non-toxic, used to evaluate detoxification models. The collection was curated from a variety of toxicity datasets (Risch et al., 2021; Demus et al., 2022; Hanu and Unitary team, 2020; Elisabetta Fersini, 2020; Nozza et al., 2023), from which 2500 toxic examples were selected for each subset. For each language subset, a group of native experts was tasked with rewriting each example, such that the toxic example becomes non-toxic with minimal changes to the example.

For toxicity in *software-engineering*, we used the

[1]We also tested and evaluated a ridge regression and a linear SVM for the linear probe, but found both consistently underperforming compared to logistic regression.

toxicity benchmark by Sarker et al. (2020), which focuses on toxicity in software-engineering discussions. This dataset is split into two subsets. First, a subset containing code reviews from three large open-source projects: Android, Chromium OS, and LibreOffice. Second, a subset containing examples from the Gitter channel of the Ethereum project.

Marks and Tegmark analyzed how LLaMA-2 models (Touvron et al., 2023) differentiate between true and false statements. We use seven datasets from this work. 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. Both the *neg_cities* and *neg_sp_en_trans* are negated versions of *cities* and *sp_en_trans*. Specifically, to each example in the two original datasets, a "not" was added and the label flipped. The *larger_than* and *smaller_than* datasets contain correct and incorrect mathematical expressions, following these structures "x is larger than y" and "x is smaller than y". Lastly, the "companies_true_false" dataset contains correct and incorrect statements about US companies.

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 were split into a training and evaluation set using an 85 / 15 split and stratified on the toxicity label to maintain class balance. We then split the training set again using an 80 / 20 for early stopping and epoch evaluation, leading to a final split (training, early stop, evaluation) of 70 / 15 / 15.

6 Activation Patching

Following recent work in causal intervention (Vig et al., 2020; Meng et al., 2022; Marks and Tegmark, 2024), we apply activation patching as a diagnostic to localize which transformer layers ℓ in an LLM M contribute to the toxicity of a statement s, and to confirm that toxicity is present as a feature in M.

To measure each layer's contribution towards predicting "toxic" or "healthy" as the next token[2], we intervene in the forward pass of M, altering intermediate representations and observing the changes to M's output Y. Following Geiger et al. (2020) and Finlayson et al. (2021) we construct two minimally different inputs: a toxic prompt p_T and a healthy prompt p_H. We construct p_H by substituting a single word in p_T Both are designed as a five-shot prompt with 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 p_H and p_T, 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, as recommended by Zhang and Nanda (2024):

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

Figure 1 exemplifies the results for Qwen3-8B, whose activation pattern is observed in all LLMs in this study, with slight nuances. Specifically, a pattern of three groups of active transformer layers emerges throughout all model sizes, indicating that the feature *toxicity* is present in each model. 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. As our target example spans over two and a half sentences, we observe this 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 Llama-3-families is the strength of the three groups**, which emerges with a higher log difference in Qwen3 than in Llama-3.

7 Measuring Non-Linearity

We base our measurement of non-linearity on the accuracy gain (or loss) of a non-linear MLP probe over a linear LR probe. Specifically, let $\mathcal{D} = \{d_1, d_2, \ldots, d_K\}$ be the ordered sequence

[2]We use "toxic" and "healthy", as both the Qwen3 and Llama-3 tokenizer encodes them as one token

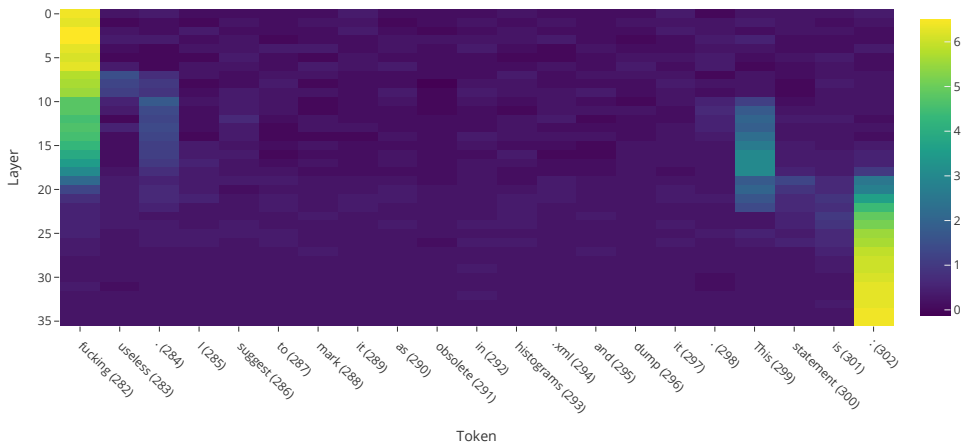


Figure 1: Activation patching for Qwen3-8B using two nearly identical few-shot prompts with one substituted word (*slightly* → *fucking*). We plot the *log difference* between tokens "toxic" and "healthy" for each token and model layer. Three groups of active layers emerge: The first over the substituted word, the second on the first word of a new sentence, and the third group over the last token.

of MLP hidden dimensions in the capacity sweep, with $d_1 < d_2 < \cdots < d_K$ (we use $\mathcal{D} = \{8, 16, 32, 64, 128, 256, 512\}$). Then $\text{acc}_{\text{MLP}}(d_k)$ denotes the accuracy of the MLP probe with hidden dimension d_k, and acc_{LR} the accuracy of the linear probe at the same configuration (model, layer, training medley, validation dataset), such that the accuracy gain (loss) for a specific MLP (g_k) is:

$$g_k = \text{acc}_{\text{MLP}}(d_k) - \text{acc}_{\text{LR}}, \qquad k = 1, \ldots, K. \tag{8}$$

Figure 2 shows the LR accuracy and MLP accuracy gain for Qwen3-4B on general toxicity, showing that the accuracy of the linear probe increases consistently over the first half of the residual stream, plateauing from layer 17 to layer 22, with a peak of 0.87 at layer 22, before decreasing steadily down to 0.8 at the final layer of the model. Here, the non-linear probes show a higher probing accuracy in earlier and later layers compared to the linear probe, while their accuracy is marginally weaker or equal for most parts between layers 15 and 23. Furthermore, the figure shows that the accuracy gain is not consistently strong (or weak) throughout the hidden dimension sweep, but increases with size in earlier layers, and fluctuates in later layers.

8 The Geometry of Toxicity Diverges Across Model Families and Domains

To analyze the non-linearity of feature representation occurring in the activation space of LLMs, we first define the mean non-linearity (NL_{mean}) of a transformer layer's representation O_ℓ as:

$$NL_{\text{mean}} = \frac{1}{K} \sum_{k=1}^{K} g_k \tag{9}$$

This measure provides an overview of the non-linearity present in M's activation space, indicating which layers show non-linearity and which do not, as well as how non-linearity evolves throughout the residual stream of M.

Figure 3a reports NL_{mean} for the Qwen3 and Llama-3 families on *general toxicity*, with layers standardized into relative model depth to enable comparison across models of varying sizes. Qwen3 shows positive NL_{mean} across all model depths, hovering around $1pp$, with a higher NL_{mean} in both lower (up to $1.8pp$) and higher relative depths (up to $2pp$). Llama-3, by contrast, shows a higher NL_{mean} only in the first third, with a peak $4.9pp$, and decreasing to near-zero thereafter. This indicates that a linear probe captures Llama-3's toxicity representation well after the early layers, whereas a non-linear component persists throughout Qwen3, especially during early encoding and late decoding. The two families thus encode toxicity with distinct geometric structures.

To determine whether this difference is domain-dependent, we also analyze NL_{mean} across multilingual (Figure 3b), and software-specific (Figure 3c) toxicity. While the strength of non-linearity varies by domain, the core pattern, higher early-layer non-linearity in Llama-3 and consistent non-linearity in Qwen3 with peaks at both ends, holds across all three types. The most notable variation occurs for software-specific toxicity, where the two

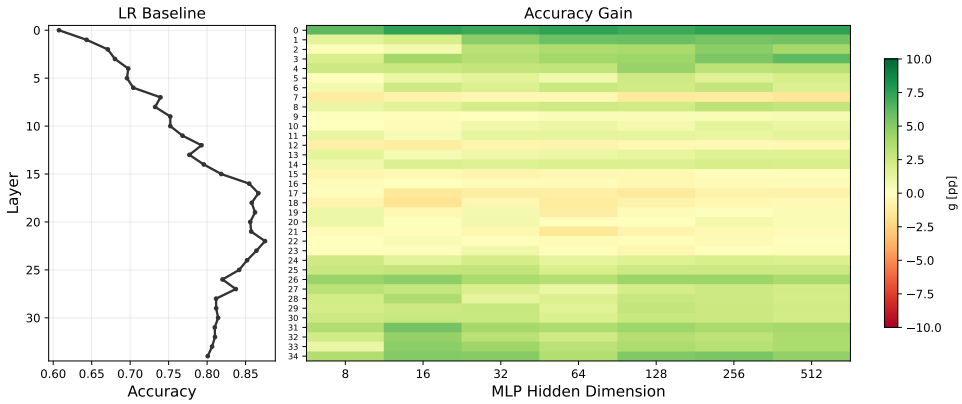


Figure 2: LR accuracy (left) and MLP accuracy gain (right) in percentage points (pp) for probing *general toxicity* in the activation space of Qwen3-4B across layers and hidden dimensions of the non-linear probes. The non-linear probes have a higher accuracy at classifying general toxicity in the first 6 and last 10 layers of Qwen3-4B, while being only marginally better in layers 7 - 14, and marginally weaker in layers 15-23. The largest accuracy gain of 7.4pp is achieved on the first layer for $d_2 = 16$.

families show a progressively greater disparity towards higher relative depths. Specifically, Llama-3's non-linear probes underperform the linear probe here on average, while Qwen3 rises to an NL_{mean} of $3.6pp$.

These results demonstrate that the general geometric differences between the two families are consistent across toxicity types, yet differences in strength and nuances to the patterns exist. The non-linear probe captures the representation of toxicity in early layers better for both the Llama-3 and Qwen3 families, as well as the representation of toxicity in the second half of the residual stream for the Qwen3 family. However, these results do not reveal how much additional information the best non-linear probe captures, nor how consistent non-linear probes are across layers.

9 Relative Improvement and Consistency of Toxic Representations

An inherent limitation of NL_{mean} is that it is not a monotone function of either the magnitude or the structural complexity of non-linearity, as different cases can yield identical means but represent different observations. Let $K = 7$, $c > 0$, and consider two layers:

$$\text{Layer A} : g_k = c \quad \forall k, \tag{10}$$

$$\text{Layer B} : g_k = 0 \quad \forall k < 7, \quad g_7 = 7c \tag{11}$$

Then $NL^A_{\text{mean}} = c = NL^B_{\text{mean}}$. Here, the accuracy gain in Layer A is fully captured by the smallest MLP ($h_1 = 8$), whereas no accuracy gain occurs until the maximum capacity is reached in Layer B. These are representationally and interpretively distinct situations, yet NL_{mean} assigns them identical values.

To improve our measure of non-linearity, we introduce the consistency-weighted strength of non-linearity (NL_C), which reports both the magnitude and consistency of non-linearity. Let $G = \max g_k \forall k \in K$, be the highest accuracy gain for a given model-layer-datasets combination (i.e., the *magnitude* of non-linearity), then

$$NL_C = G \cdot C \tag{12}$$

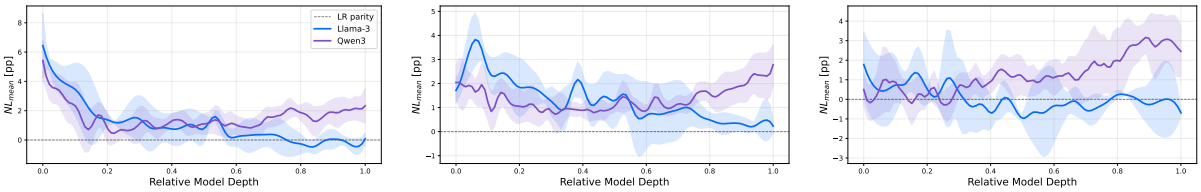
is the consistency-weighted strength of non-linearity, where C is the gain-normalized fraction of capacities at which the MLP beats the LR, defined as follows:

$$C = \frac{1}{K} \sum_{k=1}^{K} \text{clip}\left(\frac{g_k}{G}, 0, 1\right). \tag{13}$$

NL_C allows us to visualize the relationship between G and C. Specifically, it enables us to analyze whether the large accuracy gains achieved by MLP probes are backed by consistent improvement.

Figure 4 shows the relationship between G and C averaged over all toxicity datasets. Here, the relative layer depth is indicated by the transparency of the dots, and the size of the models is represented by the size of each dot. Furthermore, we standardize layers into deciles.

Llama-3 shows the highest gain and consistency for the first decile of Llama-3-1B and -8B, and the second decile of -1B, whereas most middle and late deciles show very limited non-linearity. Hence, only four deciles reach an $NL_C \geq 2$ and four early to mid deciles an $NL_C \geq 1$, supporting our previous observation that non-linearity in the representation of toxicity occurs only in the activation space of early deciles in Llama-3.



(a) General Toxicity (b) Multilingual Toxicity (c) Software-specific Toxicity

Figure 3: Mean accuracy gain NL_{mean} in percentage points, trained on the activation space of all Qwen3 and Llama-3 models, for **(a)** *general toxicity*, **(b)** *multilingual toxicity*, and **(c)** *software-specific toxicity*. We plot NL_{mean} over relative model depth (x-axis). The shaded area depicts the 95% confidence interval. Llama-3 has a high NL_{mean} in lower depths, with little to no non-linearity thereafter for general and software-specific toxicity, and continuously decreasing non-linearity for multilingual toxicity. Qwen3 shows positive NL_{mean} for all three types of toxicity, with high NL_{mean} in both low and high relative model depths, and the strongest separation between model families.

Qwen3 shows the highest gain and consistency for the latest deciles, especially for the last decile in Qwen3-1.7B, reaching $NL_C = 3.9$. Earlier deciles, as well as a few middle and late deciles, reach an NL_C between 1 and 2, and most middle deciles reach an $NL_C \leq 1$. An exception is Qwen3-30B-A3B (depicted with a black outline), which is weaker in later deciles than in the first decile.

In general, Qwen3 exhibits both higher consistency and maximum gain, with most deciles achieving $G \geq 1$ and $C \geq 0.6$. Hence, only a few deciles score $NL_C \leq 0.5$, with the lowest non-linearity score of $NL_C = 0.127$ reached by the second decile in Qwen3-32B.

These results show that for the representation of toxicity, a high consistency (C) is a necessary condition for large gains (G), but not a sufficient one. For Qwen3, for example, no decile reaches $G \geq 2$ with $C \leq 0.6$, while the decile with the highest consistency, decile 7 in Qwen3-8B, reaches a C of 0.91, while only reaching a G of 1.9. We can therefore assume that the occurrence of non-linearity in the representation of toxicity can be probed consistently using 2-layer MLPs in the activation space of LLMs.

10 Validation on Truthfulness

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.

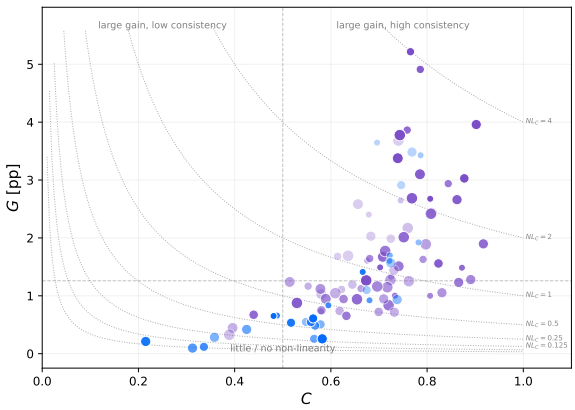


Figure 4: Consistency-weighted strength of non-linearity NL_C for *toxicity*. The transparency of the dots indicates the relative layer depth (higher transparency → lower depth), and the size of each dot represents the size of the models (larger dot → larger model). Qwen3 ● shows the highest gain and consistency for the latest deciles, while Llama-3 ● shows the highest gain and consistency for the first decile.

In Figure 5, we show that in the activation space of Qwen3 there are, on average, slight traces of non-linearity present, but no larger amounts, nor any consistent non-linearity above 1pp. For Llama-3, there is no non-linearity (except for a very slight amount around a relative depth of 0.2) present in the activation space of Llama-3. On the contrary, the non-linear probes show weaker results than the LR baseline (with a few positive outliers), beginning at a relative depth of 0.4, suggesting that Llama-3 represents features primarily linear in their activation.

This is further supported by Figure 6. While on average, only a marginal amount of non-linearity is present over the average of all Qwen3 models,

we find deciles 8 and 10 of Qwen3-1.7B, reaching an $NL_C \geq 2$, and overall 14 deciles of different Qwen3 models, reaching an $NL_C \geq 1$. Of these 14 deciles, only two deciles, namely deciles 4 and 5 of Qwen3-32B, are located in the first half of their models' residual stream, and none is located in the first third. For Llama-3, only two deciles, decile 4 of Llama-3-8B and decile 5 of Llama-3-1B, reach an $NL_C \geq 0.5$, while all other deciles of the three Llama-3 models reach an $NL_C \leq 0.5$.

These results are important for two reasons. First, they show that non-linearity can be present in highly linear features, especially in certain layers and areas of specific models. Second, they give evidence that non-linear probes are not a better classifier of features in general, but a distinct measure for finding non-linearities in the representation of certain features in specific models and layers.

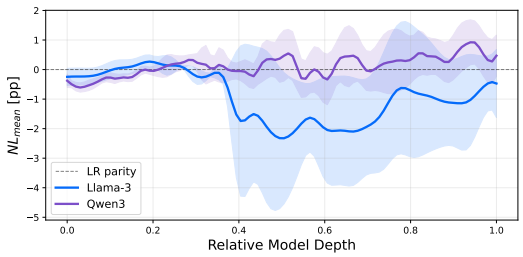


Figure 5: Mean accuracy gain NL_{mean} in percentage points for *truthfulness*. We plot NL_{mean} over relative model depth (x-axis). The shaded area depicts the 95% confidence interval. Qwen3 shows no to little non-linearity over the first two-thirds, with some non-linearity emerging in the final relative third, while the non-linear probes underperform in most parts of Llama-3.

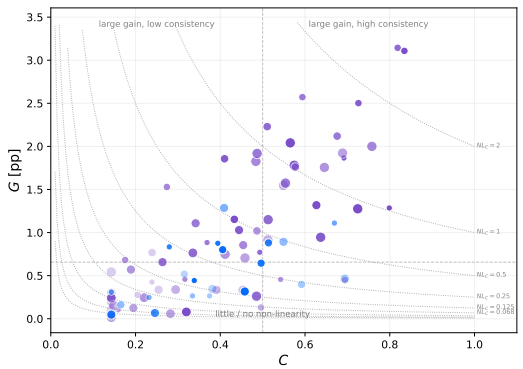


Figure 6: Consistency-weighted strength of non-linearity NL_C for *truthfulness*. The transparency of the dots indicates the relative layer depth (higher transparency → lower depth), and the size of each dot represents the size of the models (larger dot → larger model). Qwen3 ● shows both a higher strength and consistency in the emerging non-linearity than Llama-3 ●.

11 Discussion and Conclusion

In this paper, we investigated whether *toxicity* is linearly represented in Qwen3 and Llama-3, and report where non-linearities in the representation arise. To achieve this, we conducted a combination of linear and non-linear probing experiments, using non-linear probes of different capacities. For these experiments, we used nine toxicity datasets, spanning general, multilingual, and software-specific toxicity. Furthermore, we validated our findings on the feature truthfulness, using seven datasets from Marks and Tegmark (2024).

Our findings show that toxicity is not represented uniformly across LLM architectures. For Qwen3, non-linear probes consistently reached a higher probing accuracy than the linear probe across the residual stream, particularly in early and later layers. Llama-3 showed a different pattern: non-linearity is confined to the first third of the residual stream, while the remaining two-thirds were mostly linear.

Furthermore, we show that the geometric differences between Llama-3 and Qwen3 are consistent characteristics across toxicity types, but nuances arise for different types of toxicity. Specific contexts contain subtle cues (e.g., technical jargon, community norms, and implicit tone) that impact the strength of the non-linear patterns in specific areas of the residual stream. The most notable variation occurs in software-specific toxicity, where a greater disparity in non-linearity emerges between the two model families.

Finally, our experiments show that studying non-linearity is important for evaluating how well linear probing captures the representation of a feature. Non-linear probes can be used to detect if and in which areas of an LLM, non-linear patterns arise in its activation space. Furthermore, a set of non-linear probes of different capacities allows one to study how consistent the strength of the non-linear pattern is.

Future research should focus on understanding how non-linearity is added to the activation space of LLMs, and why some model families are more prone to it than others. Furthermore, researchers should determine which features exhibit larger amounts of non-linearity, to study what common drivers or underlying patterns drive the exhibition of non-linearity (e.g., the feature is hierarchical or a combination of multiple more linear features).

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 and mid-size LLMs, giving no insight into larger LLMs. Second, we don't conduct a deeper analysis into why Qwen3 models exhibit stronger non-linearity than the Llama-3 models to understand and find the differences in architecture or training, which lead (or don't lead) to the non-linearization of features in activation space. Finally, 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*.

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.

Daryna Dementieva, Daniil Moskovskiy, Nikolay Babakov, Abinew Ali Ayele, Naquee Rizwan, Frolian Schneider, Xintog Wang, Seid Muhie Yimam, Dmitry Ustalov, Elisei Stakovskii, Alisa Smirnova, Ashraf Elnagar, Animesh Mukherjee, and Alexander Panchenko. 2024. Overview of the multilingual text detoxification task at pan 2024. In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*. CEUR-WS.org.

Christoph Demus, Jonas Pitz, Mina Schütz, Nadine Probol, Melanie Siegel, and Dirk Labudde. 2022. Detox: A comprehensive dataset for German offensive language and conversation analysis. In *Proceedings of the Sixth Workshop on Online Abuse and Harms (WOAH)*, pages 143–153, Seattle, Washington (Hybrid). 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, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *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*.

Paolo Rosso Elisabetta Fersini, Debora Nozza. 2020. Ami @ evalita2020: Automatic misogyny identification. In *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*, Online. CEUR.org.

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.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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

Laura Hanu and Unitary team. 2020. Detoxify.

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.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

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

Tomas 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, Atlanta, Georgia. Association for Computational Linguistics.

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.

Debora Nozza, Alessandra Teresa Cignarella, Greta Damo, Tommaso Caselli, and Viviana Patti. 2023. HODI at EVALITA 2023: Overview of the Homotransphobia Detection in Italian Task. In *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023)*, Parma, Italy. CEUR.org.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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.

Julian Risch, Anke Stoll, Lena Wilms, and Michael Wiegand. 2021. Overview of the GermEval 2021 shared task on the identification of toxic, engaging, and fact-claiming comments. In *Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments*, pages 1–12, Duesseldorf, Germany. Association for Computational Linguistics.

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.

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, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *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 Details on Early Stopping

In §3, we define a linear and a non-linear probe, which we train using early stopping, stopping the probe training when the accuracy of the probe does not improve by 1e-4 over 20 epochs, while also training for at least 10 epochs, to ensure that each probe does not converge early in a local minimum. Following, we show the distribution of the best epoch for each probe type.

Figure 7 shows the distribution for probes trained on *toxicity* datasets.

Figure 8 shows the distribution for probes trained on *truthfulness* datasets.

Both show very similar results, with the linear probe needing, on average, the most epochs to reach its best accuracy, while also showing the highest variance. Furthermore, for both model families and features, the MLP reaches its best epoch earlier on average for each capacity increase.

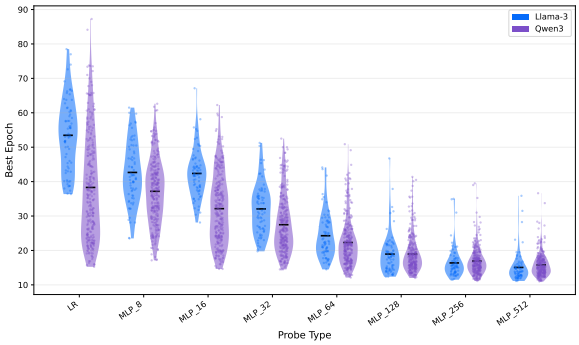


Figure 7: Distribution of the best epoch (determined by early stopping) for *toxicity* probes, grouped by probe type. The black line depicts the mean best epoch per probe and model family.

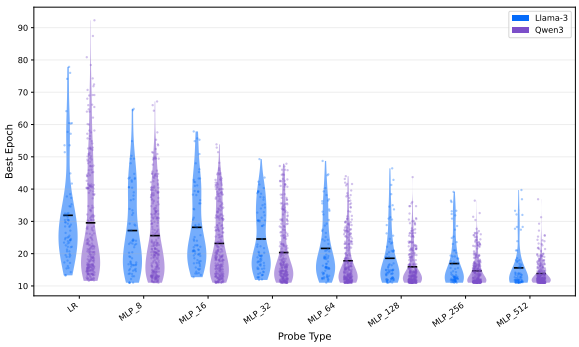


Figure 8: Distribution of the best epoch (determined by early stopping) for *truthfulness* probes, grouped by probe type. The black line depicts the mean best epoch per probe and model family.

B Model Details

In this section, we give a more detailed overview of the 10 models used in this study. Specifically, we provide details on the architectural details and hyperparameters of the models.

Table 1: Details for the Qwen3 models.

Model	Layers	D_{hidden}	Q-Heads	KV-Heads	Experts
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	✖
Qwen3-30B-A3B	48	2,048	32	4	128/8

Table 2: Details for the Llama-3 models.

Model	Layers	D_{hidden}	Q-Heads	KV-Heads	Experts
Llama-3.2-1B	16	2,048	32	8	✖
Llama-3.2-3B	28	3,072	24	8	✖
Llama-3.1-8B	32	4,096	32	8	✖