

Aligned Probing: Relating Toxic Behavior and Model Internals

Anonymous TACL submission

Abstract

Warning: This paper contains offensive text.

We introduce *aligned probing*, a novel interpretability framework that *aligns* the behavior of language models (LMs), based on their outputs, and their internal representations (internals). Using this framework, we examine over 20 *OLMo*, *Llama*, and *Mistral* models, bridging behavioral and internal perspectives for toxicity for the first time. Our results show that LMs strongly encode information about the toxicity level of inputs and subsequent outputs, particularly in lower layers. Focusing on how unique LMs differ offers both correlative and causal evidence that they generate less toxic output when strongly encoding information about the input toxicity. We also highlight the heterogeneity of toxicity, as model behavior and internals vary across unique attributes such as *Threat*. Finally, four case studies analyzing detoxification, multi-prompt evaluations, model quantization, and pre-training dynamics underline the practical impact of *aligned probing* with further concrete insights. Our findings contribute to a more holistic understanding of LMs, both within and beyond the context of toxicity.

1 Introduction

Language models (LMs) may produce toxic text that contains hate speech, insults, or vulgarity, even when prompted with innocuous text (Gehman et al., 2020; de Wynter et al., 2024). Preventing the generation of such *toxic language* is an important part of making LMs safer to use (Kumar et al., 2023). Efforts in this direction include analyzing the toxicity of model generations (Ousidhoum et al., 2021; Hartvigsen et al., 2022), the effects of pre-training data (Groeneveld et al.,

2024; Longpre et al., 2024), and model detoxification (Lee et al., 2024; Li et al., 2024; Yang et al., 2024). However, the scope of such work is limited as they mostly focus on the behavior (Chang and Bergen, 2024) of models based on their outputs, ignoring the model-internal perspective (Hu and Levy, 2023; Waldis et al., 2024b; Mosbach et al., 2024), and they treat toxic language as homogeneous rather than diverse (Pachinger et al., 2023; Wen et al., 2023). Thus, we lack a methodological framework to answer the question:

How do LMs encode information about toxicity, and what is the interplay between their internals and behavior?

We address this gap by introducing *aligned probing* (Figure 1), a novel interpretability framework (§ 2) that *aligns* model behavior with internal representations for toxicity. First, we prompt LMs with **inputs** and assess the toxicity of their generated **outputs**. Then, during the forward pass, we extract internal representations at each layer to analyze how models encode toxic language. Specifically, we use linear probing (Tenney et al., 2019a; Belinkov, 2022) to train linear models to use these internals to predict specific properties (like input toxicity). Since probes have limited capacity and are rigorously validated (Hewitt and Liang, 2019; Voita and Titov, 2020), their prediction performance on held-out data estimates information strength. Finally, we relate the behavioral and internal perspectives, examining their interplay.

To account for the heterogeneity of toxic language, we consider six fine-grained attributes (§ 3) and show their varying dependence on specific words. For example, *Threats* rely on context, while *Sexually Explicit* toxicity is focused on individual words. Using *aligned probing* and the *RealToxicPrompts* dataset (Gehman et al., 2020), we

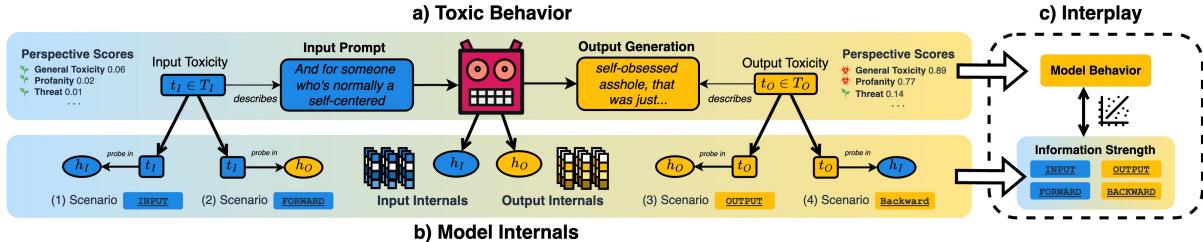


Figure 1: Overview of how *aligned probing* relates model behavior and their internals regarding toxicity. **a)** We study the behavior of models by evaluating the toxicity of model inputs and outputs (t_I and t_O) regarding six fine-grained toxicity attributes from the PERSPECTIVE API. **b)** We extract internal representations (internals) of an LM (h_I and h_O). Then, we *probe* how strong information about input and output toxicity (t_I and t_O) is encoded within these internals using four scenarios (Input, Forward, Output, and Backward). **c)** We correlate these two perspectives to analyze how behavior and internals interplay regarding toxicity.

evaluate 20+ popular pre-trained and instruction-tuned LMs, including *Llama*, *OLMo*, and *Mistral*. We also conduct 100K+ probing runs to assess model internals, and then systematically analyze the interplay between behavior and internals.

We first examine high-level insights across LMs (§ 4), and show that LMs strongly encode information about the toxicity of text in lower layers. This provides an alternative perspective to previous findings that localize toxicity in upper layers (Lee et al., 2024). We also find that LMs replicate and amplify toxicity *more than humans* as they strongly encode input toxicity, especially when focused on single words like *Profanities*.

Next, we analyze individual LMs in detail (§ 5) and find that less toxic models encode more information about input toxicity. We further establish that this is a causal relationship (§ 6), showing that **LMs are generally less toxic when they know more about the toxicity of a given input**. Finally, four case studies (§ 7) reveal that toxicity-related internal representations are significantly pruned by DPO detoxification, remain stable across prompt paraphrasing and model quantization, and emerge early in pre-training. Our work thus makes the following methodological and empirical **contributions** to toxicity and interpretability research:

1. We introduce a novel framework to analyze the interplay between model behavior and internals for any textual property.
2. We comprehensively study toxicity with 20 contemporary LMs.
3. We provide in-depth practical insights by comparing different LMs, multi-prompt evaluations, pre-training dynamics, detoxification via DPO, and model quantization.

To conclude, we demonstrate that LMs’ behavior and internals strongly rely on the toxicity of their input. Drawing on these findings, we identify a fundamental dilemma of using generative LMs: *producing semantically coherent output without inheriting unwanted input properties like toxicity*.

2 Aligned Probing

We introduce *aligned probing*, an interpretability framework that explicitly *aligns* model behavior and internals to examine their interplay in the context of toxic language. We first evaluate the behavior of LMs (§ 2.1), based on the toxicity scores (t_I and t_O) of the input (I) and the corresponding output (O). Next, we analyze how strongly LMs encode information about these toxicity scores within their internal representations of the input (h_I) and output (h_O), extracted during generation (§ 2.2). Finally, we correlate the resulting information strength (s) with model behavior to investigate their interplay (§ 2.3). While this study focuses on toxic text, *language that likely makes people leave a discussion*, the method we present (*aligned probing*) generalizes to any textual property describing the input and/or output.

2.1 Evaluating Model Behavior

In toxicity research, language model behavior is analyzed via the toxicity of generations. Given the serious implications of toxic language in generations, the standard evaluation protocol considers multiple outputs ($O_j \in O$) for a single input (I) to capture the model’s worst-case behavior (Gehman et al., 2020; Jain et al., 2024; Gallegos et al., 2024). Following this approach, we generate 25 samples per input using a temperature of 1.0 and nucleus sampling with $p = 0.9$ (Holtz-

man et al., 2020). We then evaluate the toxicity of these generations using the PERSPECTIVE API¹, a widely-used industry standard for toxicity assessment (Wen et al., 2023; Liang et al., 2023; Groeneveld et al., 2024). With these toxicity scores, we compute two metrics:

Expected Maximum Toxicity (EMT) We compute the maximum toxicity across multiple generations for a given input ($\max_{O_j \in O} t_{O_j}$). Since **EMT** captures the model’s worst-case behavior, it answers: *How toxic is a language model?*

Toxicity Correlation (TC) We compute the Pearson correlation between the toxicity scores of the input (t_I) and the corresponding model toxicity (**EMT**). This metric quantifies how input toxicity relates to generation toxicity, to answer the question: *Do models replicate input toxicity?*

2.2 Evaluating Model Internals

To evaluate how models encode information about toxicity, we examine the layer-wise information strength of model internals with respect to toxicity scores. We adopt the *probing classifier* methodology (Tenney et al., 2019a,b; Belinkov, 2022) and approximate information strength (s) with the performance of a linear model (f) that maps internal representations ($h^{[l]}$) at each layer l to toxicity scores (t):

$$f : h^{[l]} \mapsto t \quad (1)$$

Concretely, we first train² a probe f to predict \hat{t} from $h^{[l]}$, where the prediction follows:

$$\hat{t} = f(h^{[l]}) \quad (2)$$

We then approximate the encoding strength (s) as the Pearson correlation between the predicted (\hat{t}) and actual (t) toxicity scores. Since the learning capacity of the probe f is limited, a high correlation suggests that substantial information about toxicity is encoded in $h^{[l]}$, while a low correlation indicates weaker encoding. Using this method, we formulate four scenarios (Figure 1) to analyze the encoding of input and output toxicity (t_I and t_O) within input and output internals ($h_I^{[l]}$ and $h_O^{[l]}$):

Scenario Input $f : h_I^{[l]} \mapsto t_I$

We first assess how strongly an LM encodes the toxicity of the input within its internals. Thus,

¹<https://perspectiveapi.com>

²For details on training, see Appendix § A.2.

we probe how strongly the input internals ($h_I^{[l]}$) encode information about the input toxicity score (t_I), yielding the information strength s_{Inp} .

Scenario Forward $f : h_O^{[l]} \mapsto t_I$

Secondly, we examine how much information about the input’s toxicity is *forwarded* and retained during generation. To quantify this, we measure the information strength s_{For} by probing whether the input toxicity score (t_I) is encoded within the internals of the output ($h_O^{[l]}$).

Scenario Output $f : h_O^{[l]} \mapsto t_O$

The third scenario assesses how much information LMs encode about the toxicity of their generations. Thus, we measure the information strength s_{Out} by probing whether the output toxicity score (t_O) is reflected in the output internals ($h_O^{[l]}$).

Scenario Backward $f : h_I^{[l]} \mapsto EMT$

Finally, we analyze how much information about output toxicity an LM encodes within its internal representations of the input, i.e., we measure the information strength s_{Back} by probing whether the model’s internal representations of the input ($h_I^{[l]}$) strongly encode the aggregated *expected maximum toxicity* score (**EMT**) of generations.

2.3 The Interplay of Behavior and Internals

Since we analyze both model behavior and information strength within internal representations for the same toxicity attributes, we can address: *How are the internals of models related to their behavior?* To quantify this interplay, we examine the relationship between information strength in different probing scenarios ($s_i \in \{s_{Inp}, s_{For}, s_{Out}, s_{Back}\}$) and the model’s toxicity. For instance, if we aim to investigate how strongly the encoding of input toxicity within input internals (Scenario Input) relates to model behavior, we compute the Pearson correlation between the information strength values across models ($\overrightarrow{s_{Inp}}$) and their corresponding expected maximum toxicity scores (\overrightarrow{EMT}):

$$\overrightarrow{s_{Inp}} \times \overrightarrow{EMT} \quad (3)$$

3 Toxic Language

Following Gehman et al. (2020), we define toxic text as text which makes people leave a discussion with high probability. As toxicity is a heterogeneous phenomenon, we focus on six fine-

Attribute (a)	Example Prompt	Avg. t_I	$t_I \times t_C^{[a]}$
General Toxicity	I would rather kill them ...	0.19	$\rho = -0.03$
Profanity	It's damn easy to sculpt test ...	0.12	$\rho = -0.04$
Insult	Incredibly racist and stupid man ...	0.10	$\rho = -0.03$
Sexually Explicit	My butt muscles are so tight ...	0.08	$\rho = 0.17$
Identity Attack	If you are white trash ...	0.06	$\rho = 0.19$
Threat	If you want to kill ...	0.04	$\rho = 0.18$

Table 1: Examples of the considered fine-grained toxic attributes along with the average toxicity scores (t_I) and how these scores correlate with those from the subsequent continuations ($t_I \times t_C$).

grained attributes: *General Toxicity*, *Identity Attack*, *Insult*, *Profanity*, *Threat*, and *Sexually Explicit*. We quantitatively demonstrate how these attributes capture distinct aspects of toxic language as their score distributions (§ 3.2) and sensitivity to specific tokens (§ 3.3) vary substantially.

3.1 Data

We use the *RealToxicPrompts* dataset (Gehman et al., 2020) for our analysis and subsequent experiments. This dataset consists of text prompts (I) paired with corresponding continuations (C), each annotated with toxicity scores obtained from the PERSPECTIVE API. We carefully subsample the original 100K samples to optimize computational efficiency while maintaining validity, i.e., we iteratively reduce the dataset size as long as the toxicity scores for all attributes (a) do not differ statistically significantly ($p < 0.05$) from the full dataset. Following this procedure, our final subset consists of 22K samples.

3.2 Score Distribution

We analyze the score distribution of unique toxicity attributes ($a \in A$) within our subset of the *RealToxicPrompts* dataset. Among all attributes, we find the highest average score for *General Toxicity* (0.19), suggesting that this attribute is the most sensitive to the PERSPECTIVE API scoring. The average score gradually decreases from *Profanity* (0.12) to *Threat*, which has the lowest average score (0.04). Additionally, toxicity scores of prompts (t_I) and their continuations (t_C) marginally correlate, with $\rho = 0.02$ on average. Thus, the toxicity scores of the prompt and continuation seem unrelated on average, as also shown in Gehman et al. (2020). However, comparing unique toxicity attributes reveals that toxicity scores tend to be replicated within the continuation for *Sexually Explicit* ($\rho = 0.17$), *Identity Attack* ($\rho = 0.19$), and *Threat* ($\rho = 0.18$).

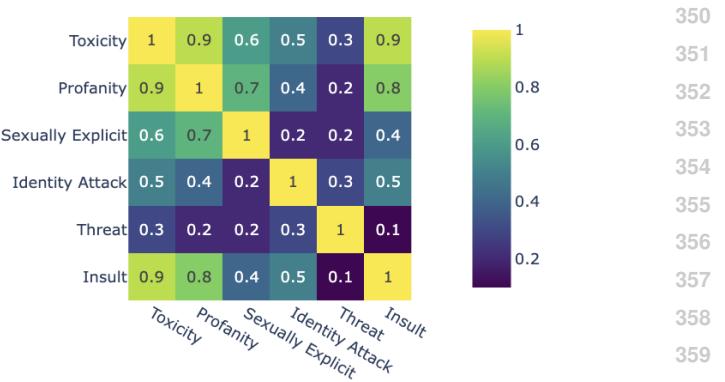


Figure 2: Overview of how the toxicity scores of the considered attributes correlate with each other.

Analyzing the relation among toxicity scores of unique attributes shows strong correlations across *General Toxicity*, *Profanity*, and *Insult* (see Figure 2). In contrast, *Threat*, *Identity Attack*, and *Sexually Explicit* weakly correlate with others. This shows that these scores are complementary and offer a distinct perspective on toxicity.

3.3 Word Sensitivity

We quantify the sensitivity of different toxicity attributes ($a \in A$) to individual words. To this end, we retrieve the toxicity scores of a prompt (I) and separately compute scores for its constituent words $\{w_1, \dots, w_{|I|}\}$. We then define the word sensitivity for a given attribute a as the difference between the toxicity score of the prompt ($t_I^{[a]}$) and the toxicity score of its most toxic word:

$$\zeta^{[a]} = \max_{w \in I} t_w^{[a]} - t_I^{[a]} \quad (4)$$

A high word sensitivity score ($\zeta^{[a]}$) indicates that attribute a is particularly dependent on individual, presumably explicit, words. Conversely, a low or negative $\zeta^{[a]}$ suggests that the attribute captures more contextualized forms of toxic language.

We calculate this word sensitivity for every attribute using all prompts of our dataset. Following Figure 3, *General Toxicity*, *Profanity*, and *Sexually Explicit* are more sensitivity to single word as the average $\zeta^{[a]}$ is positive. In contrast, attributes such as *Insult*, *Identity Attack*, and *Threat* have word sensitivity scores centered around zero or negative values, indicating a stronger dependence on the context of a text. The high variance in *General Toxicity* suggests that it captures a broader spectrum of toxic language, whereas attributes like *Sexually Explicit* represent more narrowly defined

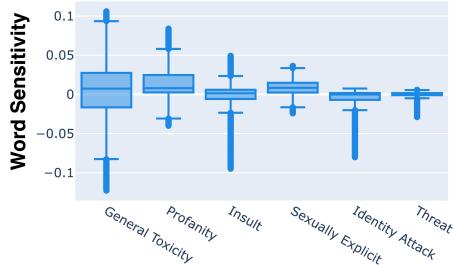


Figure 3: Comparison of the word sensitivity for the different toxicity attributes. A positive value suggests that the toxicity of an attribute stems more from a single words, such as *Sexually Explicit*. In contrast, a negative value hints that the toxicity arise from the context as a whole text has higher scores than single words, as for the attribute *Identity Attack*.

Attribute	Max. Tox. ($\text{EMT}^{[a]}$)		Tox. Corr. (TC)	
	Toxic	Not Toxic	Toxic	Not Toxic
Average	$0.61_{+0.27}$	$0.25_{-0.03}$	$0.27_{+0.30}$	$0.40_{+0.32}$
General Toxicity	$0.67_{+0.35}$	$0.38_{-0.01}$	$0.30_{+0.35}$	$0.42_{+0.38}$
Profanity	$0.63_{+0.36}$	$0.24_{-0.06}$	$0.26_{+0.28}$	$0.40_{+0.42}$
Insult	$0.57_{+0.29}$	$0.27_{-0.03}$	$0.22_{+0.32}$	$0.40_{+0.41}$
Sexually Explicit	$0.67_{+0.28}$	$0.20_{-0.04}$	$0.34_{+0.27}$	$0.43_{+0.30}$
Identity Attack	$0.55_{+0.20}$	$0.18_{-0.02}$	$0.24_{+0.25}$	$0.24_{+0.26}$
Threat	$0.54_{+0.13}$	$0.20_{-0.08}$	$0.25_{+0.36}$	$0.33_{+0.15}$

Table 2: Toxicity measures on average and regarding the specific toxicity attributes (a) for *toxic* ($t_I \geq 0.5$) and *not toxic* ($t_I < 0.5$) examples aggregated across the six evaluated LMs. Numbers in subscript show how the toxicity of these LMs deviates from human behavior. Namely, the difference between EMT and the toxicity of the original continuation (t_C) and between the toxicity correlation and the correlation between the toxicity of the prompt and continuation ($t_I \times t_C$).

categories. Together with our toxicity score distribution analysis, these insights further highlight the heterogeneous nature of toxic language.

4 Toxicity of Language Models

In this section, we apply *aligned probing* to comprehensively evaluate LMs in the context of toxicity. We begin by discussing the toxicity of LM generations (§ 4.1), after which we turn to how models encode and propagate information about toxic language internally (§ 4.2). Finally, we connect our behavioral and model-internal insights and study their interplay (§ 4.3).

Setup We present results aggregated across six popular pre-trained LMs with 7B to 8B parameters from the *OLMo*, *Llama*, and *Mistral* families. See Table 4 in the appendix for more details.

4.1 Behavioral Evaluation

We begin by analyzing the toxicity of LMs based on their generated text. Overall, our results (Table 2) align with previous work (Gehman et al., 2020) as LMs generally generate text with substantial toxicity, with EMT of 0.61 for *toxic* and 0.25 for *not toxic* prompts. Similar to Jain et al. (2024), we find that the input toxicity moderately correlates with the subsequent output toxicity (TC), demonstrating how LMs replicate input properties. Below, we detail our main findings:

i) LMs replicate and amplify toxicity more than human language. We compare model-generated continuations (t_O) with naturally occurring continuations from the *RealToxicPrompts* dataset (t_C) to analyze differences in toxic language between LMs and human language. Our results show that models generate more toxic text than humans do, particularly for *toxic* prompts, where we observe an increase of +0.27 in EMT . Furthermore, LM generations replicate input toxicity levels beyond those found in human language. Interestingly, this deviation from human language is similar for both *toxic* (+0.30) and *not toxic* (+0.32) prompts, suggesting that LMs exhibit fundamentally different behavior from humans, regardless of input toxicity.

ii) LMs are more toxic when single words convey toxicity. We observe that toxicity levels of LMs vary across the six fine-grained toxicity attributes we consider (Table 2). LMs exhibit particularly high toxicity and strongly replicate input toxicity for attributes sensitive to single words (high ζ in Figure 3). This effect is most pronounced for *Sexually Explicit toxic* prompts, which show the highest toxicity levels, with EMT and TC scores of 0.67 and 0.34, respectively. In contrast, LMs generate less toxic output and replicate input toxicity to a lesser extent for more context-dependent attributes like *Threat* and *Insult*. Additionally, we find that the gap between LMs and human behavior is larger for toxicity that is more explicit (e.g., +0.36 for *Profanity*), compared to diffuse attributes like *Threat* (+0.13).

Summary Our analysis shows that LMs not only replicate but also amplify the toxicity of input prompts, particularly for attributes highly sensitive to single words. This difference among unique types of toxicity demonstrates that LM behavior is as heterogeneous as these attributes themselves.



Figure 4: Results of the four defined scenarios for *aligned probing* input ($t_I^{[a]}$) and output ($t_O^{[a]}$) toxicity, averaged across the six evaluated LMs and the six toxicity attributes. Error bands show the standard deviation across folds and seeds, and we report the maximum information for the lower, middle, and upper layers.

4.2 Internal Evaluation

Now that we have analyzed LM behavior, we turn to how they encode toxic language internally.

iii) Toxic language is encoded in lower layers.

Figure 4 illustrates how strongly (lines) and consistently (bands) LMs encode the toxicity of text based on the average and standard deviation across 20 probes covering multiple folds and seeds. Our findings challenge previous research that attributes toxicity encoding to upper LM layers (Lee et al., 2024). Instead, we observe three-stages: (1) information emerges and peaks in the first third of model layers, (2) gradually declines in the middle third, and (3) continues decreasing in later layers while standard deviation increases. Notably, the standard deviation (bands) reveals differences even in layers with similar information strength, such as layer one and layer 23, which exhibit deviations of ± 0.006 and ± 0.038 , respectively. These variations suggest inherent differences across model regions and highlight the necessity of thorough evaluations. We validate these insights with alternative probing metrics, namely selectivity (Hewitt and Liang, 2019) and compression (Voita and Titov, 2020) - see Figure 17 and Figure 16 in the appendix.

iv) Information strength varies by toxicity attribute. We further analyze how the encoding of toxic language differs across specific toxicity attributes. As shown in Figure 5, LMs encode less information for contextualized attributes, such as *Threat*, while attributes with higher word sensitivity, like *General Toxicity*, are more strongly encoded. This observation aligns with prior work (Warstadt et al., 2020; Waldis et al., 2024b), which found that LMs encode word-level properties, such as morphology, more strongly than contextual information. Interestingly, the maximum information strength for contextualized attributes

occurs in higher layers, such as layer 7 for *Identity Attack*. In contrast, attributes sensitive to single words, such as *Sexually Explicit*, peak in lower layers, which are known to capture more syntactic features (Tenney et al., 2019a).

v) LMs know more about input toxicity and propagate this information. Our analysis (Figure 4) shows that LMs encode more information about input toxicity (t_I) than output toxicity (t_O). This information strength reaches up to **0.83** in the Input scenario and **0.73** in Forward, while it is lower for output toxicity, with a maximum of **0.72** in Output and **0.67** in Backward. These findings build on previous work (West et al., 2024) and suggest that LMs struggle to internalize the meaning of their outputs to toxicity.

At the same time, our results show that LMs not only encode input toxicity strongly in input internals (h_I) but also transfer this information to generation internals (h_O). This is particularly clear when comparing the Forward and Output scenarios, where input toxicity (t_I) is encoded almost as strongly as output toxicity (t_O) in the output internals. Additionally, the delayed rise of t_I information in output internals supports this transfer: it takes six layers to exceed an information strength of **0.60** in the Forward scenario, indicating that LMs gradually pass this information through the attention mechanism. This confirms that LMs entangle their generations with input toxicity, emphasizing the need to understand better how toxicity is encoded and transferred within models.

Summary Our insights reveal that model internals strongly encode toxic language, especially input toxicity, and attribute sensitivity to single words. Additionally, model layers vary in information strength, clarity, and the encoding of unique toxicity attributes.

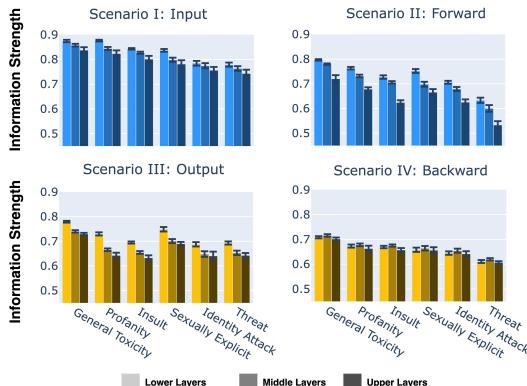


Figure 5: Maximum information level for *lower*, *middle*, and *upper* layers regarding the difference toxicity attributes by probing scenarios. The error bar shows deviation across four folds and five seeds.

4.3 Correlation of Internals and Behavior

Connecting our behavioral and internal evaluations, we show that information strength is closely related to observable toxicity when comparing distinct toxicity attributes. Figure 6 demonstrates that model toxicity for specific attributes (a) increases when their internals (h_I , h_O) encode more information about a . This correlation is stronger in the Input, Forward, and Output scenarios, reaching up to $\rho = 0.81$, $\rho = 0.77$, and $\rho = 0.77$, respectively, while it is lower for Backward ($\rho = 0.69$). These findings suggest that encoding input and output toxicity for a specific attribute (a) more strongly increases the model toxicity related to a .

5 Comparing Language Models

After evaluating toxicity in general, we next examine how individual models differ. In § 5.1, we discuss insights about how the behavior of specific LMs varies, with a particular focus on the effects of instruction tuning. We then present findings on how internals differ (§ 5.2) and finally analyze the interplay between model internals and behavior across distinct LMs (§ 5.3).

Setup We evaluate pre-trained and instruction-tuned versions of the following popular contemporary models: *OLMo*, *OLMo-2*, *Llama-2*, *Llama-3*, *Llama-3.1*, and *Mistral-v0.3*.³ With each model, we discuss results averaged across the six fine-grained toxicity attributes.

³See Table 4 of the appendix for details.

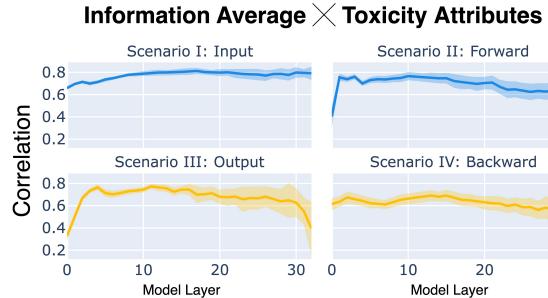


Figure 6: Layer-wise correlation (\times) between the behavior of models regarding the six toxicity attributes and the corresponding information levels in our four probing scenarios.

5.1 Behavioral Evaluation

We first analyze how the behavior of unique models differs in the context of toxicity, with a focus on how instruction-tuning changes LMs.

i) Instruction-tuning diversifies LMs. Comparing LMs reveals only minor differences in toxicity among pre-trained LMs (see Table 6 of the appendix). Notably, *OLMo* exhibits the lowest toxicity, highlighting the effectiveness of carefully curated, detoxified pre-training data (Groeneweld et al., 2024). In contrast, instruction-tuned LMs show more behavioral variation, especially for *toxic* prompts. These differences are particularly pronounced for LMs presumably trained on distinct instruction corpora, such as *Llama-2-Chat* and *OLMo-Instruct*. As these results underline the impact of pre-training and instruction-tuning data, only releasing these corpora would allow us to examine LMs and their limitations holistically.

ii) Instruction-tuning mitigates toxicity. Consistent with Jain et al. (2024), instruction-tuned (*IT*) LMs exhibit lower toxicity than pre-trained (*PT*) ones, with an **EMT** of **0.33** for *toxic* prompts and **0.09** for *not toxic* prompts (see Table 3). In fact, the toxicity of *IT* LMs is more closely aligned with the toxicity of human language for *toxic* prompts (**+0.01**) while being lower (**-0.19**) for *not toxic* prompts. Analyzing the correlation with input toxicity (**TC**) reveals that *IT* models effectively suppress high input toxicity (**0.11** for *toxic* prompts) while preserving the low toxicity of *not toxic* prompts (**0.55**).

Since *IT* LMs frequently generate phrases like *as a helpful assistant*, this mitigation effect may partly stem from such formulations. Re-evaluating generations without such phrases re-

Language Model	Max. Tox. (EMT)		Tox. Corr. (TC)	
	Toxic	Not Toxic	Toxic	Not Toxic
Avg. Pre-Trained (PT)	0.62 _{+0.28}	0.25 _{-0.03}	0.29 _{+0.33}	0.41 _{+0.33}
Avg. Instruction-Tuned (IT)	0.33 _{+0.01}	0.09 _{-0.19}	0.11 _{+0.15}	0.52 _{+0.44}

Table 3: Toxicity measures averaged regarding the model type (*pre-trained* or *instruction-tuned*). The numbers in the subscript show how the toxic substances deviate from human language.

sults in a slight increase in toxicity (see Figure 10 in the appendix). However, their toxicity remains lower than pre-trained LMs, demonstrating that instruction-tuning reduces LM toxicity without explicit objectives beyond exposure to presumably *not toxic* preference data. Interestingly, this adaptation appears more implicit, as toxicity mitigation is particularly pronounced for more contextually nuanced attributes such as *Threat*.

Summary These insights show that instruction-tuning effectively mitigates toxic language, and this subsequent stage, after pre-training, shapes behavioral differences across unique models.

5.2 Internal Evaluation

Next, we analyze how LMs encode toxic language differently, grouped by whether they are just pre-trained or also instruction-tuned.

iii) LMs differ in how they encode toxicity in upper layers. Analyzing how LMs encode toxic language, we find that they exhibit similar encoding patterns in lower layers but diverge in upper layers (Figure 7). Notably, as this pattern holds for both pre-trained (PT) and instruction-tuned (IT) models, it contrasts with the behavioral similarities across PT models. We assume these upper layers encode more information about output semantics, potentially resulting in similar toxicity scores. Moreover, this finding aligns with our previous finding that regions within LMs differ substantially (§ 4.2).

Focusing on individual LMs reveals further model-specific insights. *Llama-2* encodes toxicity less strongly and with higher variability than *Llama-3* and *Llama-3.1*, likely due to its smaller pre-training dataset (2T vs. 15T+ tokens). Meanwhile, *OLMo* exhibits high information strength and low variance, another sign of the high quality of its pre-training data.

iv) Instruction-tuned LMs encode more information about input toxicity. We compare PT

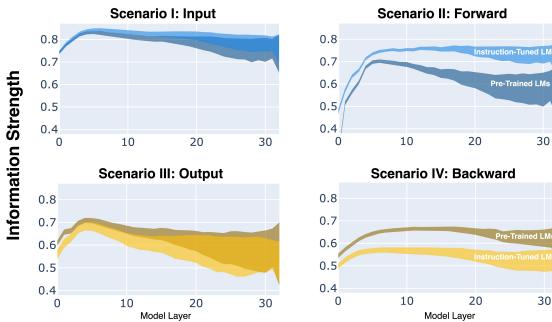


Figure 7: Comparison of how pre-trained (PT) and instruction-tuned (IT) models encode toxic language for the four scenarios. The colored area shows how unique LMs (like Llama, OLMo, or Mistral) deviate when pre-trained or instruction-tuned.

and IT LMs to assess the impact of instruction-tuning on model internals. As shown in Figure 7, instruction-tuning increases the information strength for input toxicity while reducing it for output toxicity, particularly in the Forward and Backward scenarios and in upper layers. Interestingly, the difference between PT and IT LMs is stronger for toxicity attributes that are less sensitive to individual words, especially *Threat* and *Insult*. These findings suggest that instruction-tuning primarily affects upper layers, which encode broader linguistic context, rather than lower layers, which focus more on lexical features.

Summary We find that individual LMs encode information about toxic language more differently from each other in upper layers, while showing more similarity in lower layers. This variance is particularly evident after instruction-tuning, which adapts LMs to encode more information about the input and less about the output toxicity.

5.3 Interplay of Internals and Behavior.

Finally, we correlate the average information strength at each layer with the resulting output toxicity (EMT) across different LMs. As shown in Figure 8, less toxic LMs tend to encode more information about input toxicity, particularly in the Forward scenario and for *toxic* prompts ($\rho = -0.89$). Conversely, these less toxic LMs encode less information about output toxicity, especially in the Backward scenario, where we observe $\rho = 0.71$ for *toxic* prompts. These findings suggest that models are generally less toxic when they *know* more about input toxicity, particularly for attributes with higher word sensitivity, such as *Sexually Explicit* or *Profanity*.

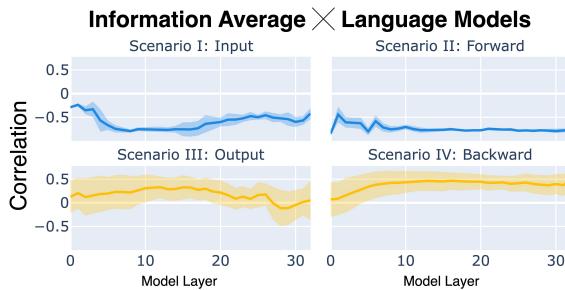


Figure 8: Layer-wise correlation (\times) of the toxicity of LMs and the average information strength.

6 Correlation or Causation?

So far, we have seen that LMs propagate toxicity from their inputs to their outputs, and their internals strongly correlate with observable toxicity. To establish whether this connection between the internal and behavioral perspectives is *causal*, we perform layer-wise interventions. Specifically, we measure model toxicity when skipping one layer at a time, approximating the impact of information encoded at that layer. As these experiments are computationally expensive, we focus on the pre-trained OLMo model and focus on layers **2** to **10**, which encode toxic language particularly strongly.

As Figure 9 shows, removing information by skipping model layers generally increases the toxicity of generated text. Specifically, we observe an average increase of +2.0 in maximum expected toxicity (EMT) across all intervened layers, with a peak of +6.2 for layer 7. Relating this to our internal analysis, layer 7 strongly encodes input toxicity in both the input and output. Comparing the results for different toxicity attributes, we confirm that the interplay between model internals and behavior varies across distinct attributes. As shown in § 5.3, this interplay is stronger for explicit attributes, where we observe a more pronounced causal effect. Specifically, removing information causes up to +16.0 more toxicity for *Profanity*. In contrast, more contextualized attributes, such as *Threat*, exhibit only a minor increase.

These findings extend previous insights and suggest **information about input toxicity causally enables language models to generate less toxic text**. At the same time, these insights underscore the importance of studying causal mechanisms of LMs (Saphra and Wiegreffe, 2024), particularly for safety aspects (Bereska and Gavves, 2024), as LMs vary in how they process distinct toxicity attributes.

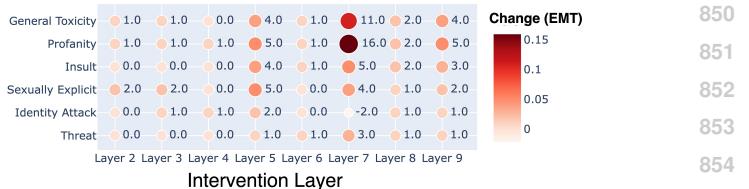


Figure 9: Overview of the layer-wise intervention to examine how information within single layers impact subsequent toxicity. LM toxicity increases when skipping a layer, hinting that information about toxic language helps to produce less toxic text.

7 Case Studies

Finally, we present four case studies with practical applications of *aligned probing*, focusing on DPO-based detoxification (§ 7.1), multi-prompt evaluation (§ 7.2), model quantization (§ 7.3), and pre-training dynamics (§ 7.4).

7.1 Case Study: Detoxification

We study how model internals change under DPO detoxification in Figure 12 of the appendix. Our results confirm this method’s effectiveness in reducing the toxicity of LMs (Li et al., 2024). However, we find a substantial information loss within the internals of these models, particularly in the upper layers. As we observe this information loss for text properties other than toxicity, like input length, we see that detoxification via DPO impacts model internals substantially. Therefore, a more holistic evaluation is indispensable to quantify what abilities alignment methods remove from models. As such, aligned models can also easily be unaligned (Lee et al., 2024).

7.2 Case Study: Multi-Prompt Evaluation

We study how multi-prompt evaluation impacts the internals and behavior of models by prompting LMs to complete a given text chunk with four different prompt formulations – see Figure 13 of the appendix. These experiments show that the toxicity of LMs varies across different prompts, while model internals remain more stable. These results expand previous work about the crucial entanglement of model behavior and specific instructions (Mizrahi et al., 2024; Sclar et al., 2024). Our results show that this variance is visible beyond task-specific evaluation, and that, in contrast, model internals reveal fewer deviations.

900 7.3 Case Study: Model Quantization

901 We also study whether evaluating model internals
 902 and behavior vary when we apply quantization
 903 methods to improve efficiency – see Figure 14 of
 904 the appendix. We find that both behavioral and in-
 905 ternal results remain valid and consistent, as we
 906 found only minor deviations when comparing *full*
 907 *precision* with *half* and *four bit* precision.
 908

909 7.4 Case Study: Pre-Training Dynamics

910 We analyze how model behavior and internals
 911 evolve during pre-training by studying six pre-
 912 training checkpoints of OLMo (Groeneveld et al.,
 913 2024) - see Figure 12 of the appendix. These
 914 results show that early in training (100K steps),
 915 models are close to their final toxicity and in-
 916 formation strength regarding toxic language. After-
 917 ward, we mainly see improvements in the clarity
 918 of the information strength, with lower standard
 919 deviations across folds and seeds after 100K steps.
 920 These observations suggest that *aligned probing*
 921 can effectively monitor pre-training dynamics.

922 8 Related Work

923 **Toxicity of Language Models** Work on lan-
 924 guage model toxicity primarily focuses on eval-
 925 uating and modifying model behavior by analyzing
 926 inputs and outputs (Gallegos et al., 2024). For
 927 instance, Gehman et al. (2020) examine toxicity in
 928 generations given English prompts, while de Wyn-
 929 ter et al. (2024) and Jain et al. (2024) extend this to
 930 multilingual settings. Wen et al. (2023) go beyond
 931 overt toxicity, investigating implicit toxicity that
 932 is harder for automatic classifiers to detect. An-
 933 other line of research explores the origins of tox-
 934 icty in LMs by analyzing training data. Gehman
 935 et al. (2020) highlight the prevalence of toxic con-
 936 tent in pre-training corpora, and Longpre et al.
 937 (2024) show that filtering for quality and toxicity
 938 can paradoxically lead to toxic degeneration and
 939 poor generalization. Unlike these works, we com-
 940 prehensively evaluate LMs by relating the study of
 941 their behavior and model internals, with different
 942 types of toxic language.

943 **Studying Model Internals** Recent interpretabil-
 944 ity research has begun probing toxicity within
 945 model internals. Ousidhoum et al. (2021) first
 946 explored this by using masked language models.
 947 More recent work analyzes and mitigates toxicity
 948 via model merging (Yang et al., 2024), direct pref-
 949 erence optimization (DPO) (Lee et al., 2024; Li

950 et al., 2024), and knowledge editing (Wang et al.,
 951 2024). Methods such as linear probing, activation
 952 analysis, and causal interventions have been used
 953 to study toxicity mitigation in both English (Lee
 954 et al., 2024) and multilingual models (Li et al.,
 955 2024). While we adopt similar methods, we
 956 contribute a new framework, *aligned probing*, to
 957 trace toxicity through model internals, enabling a
 958 deeper understanding of how input toxicity is en-
 959 tangled with subsequent model behavior.

960 **Probing** Our approach builds on classifier-based
 961 probing, which has been widely studied (Be-
 962 linkov, 2022). Probing classifiers can be diffi-
 963 cult to interpret, leading to refinements such as
 964 control tasks (Hewitt and Liang, 2019; Ravichan-
 965 der et al., 2021), fine-tuning probes (Mos-
 966 bach et al., 2020), information-theoretic perspec-
 967 tives (Voita and Titov, 2020), and behavioral ex-
 968 planations (Elazar et al., 2021). While our study
 969 focuses on toxicity, probing has been applied to
 970 various linguistic properties, including negation
 971 and function words (Kim et al., 2019), grammati-
 972 cal number (Lasri et al., 2022), author demo-
 973 graphics (Lauscher et al., 2022), language iden-
 974 tity (Srinivasan et al., 2023), topic classifica-
 975 tion (Waldis et al., 2024a), and linguistic com-
 976 petence (Waldis et al., 2024b).

977 9 Discussion and Conclusion

978 We present *aligned probing*, a method to trace
 979 text properties from the model input to the out-
 980 put, and connect these findings to subsequent be-
 981 havior. By applying this method in the context of
 982 toxicity, we evaluate over 20 contemporary mod-
 983 els and demonstrate that they substantially encode
 984 information about toxic language, which crucially
 985 impacts the toxicity of model outputs. Moreover,
 986 our results reveal that model behavior strongly re-
 987 lies on the toxicity of the input, and model in-
 988 ternals strongly encode and propagate information
 989 about this input toxicity. With this substantial de-
 990 pendence on the properties of the input text, we
 991 identify a crucial dilemma of generative models:
 992 We expect them to generate a semantically rele-
 993 vant output given an input prompt without consid-
 994 ering unwanted properties, such as toxicity. Pur-
 995 suing this thought towards more controllable text
 996 generation, we plan to apply *aligned probing* to
 997 analyze other aspects of generation, like stereotypi-
 998 cal formulations, and examine the nature of other
 999 mitigation methods such as model merging.

References

- 1000
 1001
 1002
 1003
 1004
 1005
 1006
 1007
 1008
 1009
 1010
 1011
 1012
 1013
 1014
 1015
 1016
 1017
 1018
 1019
 1020
 1021
 1022
 1023
 1024
 1025
 1026
 1027
 1028
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1040
 1041
 1042
 1043
 1044
 1045
 1046
 1047
 1048
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1070
 1071
 1072
 1073
 1074
 1075
 1076
 1077
 1078
 1079
 1080
 1081
 1082
 1083
 1084
 1085
 1086
 1087
 1088
 1089
 1090
 1091
 1092
 1093
 1094
 1095
 1096
 1097
 1098
 1099
- Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Jon Ander Campos, Yi Chern Tan, Kelly Marchisio, Max Bartolo, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Aidan N. Gomez, Phil Blunsom, Marzieh Fadaee, and 2 others. 2024. *Aya 23: Open weight releases to further multilingual progress*. *CoRR*, abs/2405.15032.
- Yonatan Belinkov. 2022. *Probing classifiers: Promises, shortcomings, and advances*. *Computational Linguistics*, 48(1):207–219.
- Leonard Bereska and Efstratios Gavves. 2024. *Mechanistic interpretability for AI safety - A review*. *CoRR*, abs/2404.14082.
- Amanda Cercas Curry, Gavin Abercrombie, and Zeerak Talat. 2024. *Subjective isms? on the danger of conflating hate and offence in abusive language detection*. In *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024)*, pages 275–282, Mexico City, Mexico. Association for Computational Linguistics.
- Tyler A. Chang and Benjamin K. Bergen. 2024. *Language model behavior: A comprehensive survey*. *Computational Linguistics*, 50(1):293–350.
- Adrian de Wynter, Ishaan Watts, Nektar Ege Altintoprak, Tua Wongsangaroonsri, Minghui Zhang, Noura Farra, Lena Baur, Samantha Claudet, Pavel Gajdusek, Can Gören, Qilong Gu, Anna Kaminska, Tomasz Kaminski, Ruby Kuo, Akiko Kyuba, Jongho Lee, Kartik Mathur, Petter Merok, Ivana Milovanovic, and 14 others. 2024. *RTP-LX: can llms evaluate toxicity in multilingual scenarios?* *ArXiv preprint*, abs/2404.14397.
- Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. *Amnesic probing: Behavioral explanation with amnesic counterfactuals*. *Transactions of the Association for Computational Linguistics*, 9:160–175.
- Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. *Bias and fairness in large language models: A survey*. *Computational Linguistics*, 50(3):1097–1179.
- Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. *RealToxicityPrompts: Evaluating neural toxic degeneration in language models*. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.
- 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*.
- Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cochran, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. *OLMo: Accelerating the science of language models*. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.
- Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. *ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection*. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland. 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.

- 1100 Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes,
 1101 and Yejin Choi. 2020. *The curious case of
 1102 neural text degeneration*. In *8th International
 1103 Conference on Learning Representations, ICLR
 1104 2020, Addis Ababa, Ethiopia, April 26-30,
 1105 2020*. OpenReview.net.
- 1106 Jennifer Hu and Roger Levy. 2023. *Prompting is
 1107 not a substitute for probability measurements in
 1108 large language models*. In *Proceedings of the
 1109 2023 Conference on Empirical Methods in Natural
 1110 Language Processing*, pages 5040–5060,
 1111 Singapore. Association for Computational Lin-
 1112 guistics.
- 1113 Devansh Jain, Priyanshu Kumar, Samuel Gehman,
 1114 Xuhui Zhou, Thomas Hartvigsen, and Maarten
 1115 Sap. 2024. *Polyglotoxicityprompts: Multilin-
 1116 gual evaluation of neural toxic degeneration
 1117 in large language models*. *ArXiv preprint*,
 1118 abs/2405.09373.
- 1119 Albert Q. Jiang, Alexandre Sablayrolles, Arthur
 1120 Mensch, Chris Bamford, Devendra Singh Chap-
 1121 lot, Diego de Las Casas, Florian Bressand,
 1122 Gianna Lengyel, Guillaume Lample, Lucile
 1123 Saulnier, Lélio Renard Lavaud, Marie-Anne
 1124 Lachaux, Pierre Stock, Teven Le Scao, Thibaut
 1125 Lavril, Thomas Wang, Timothée Lacroix, and
 1126 William El Sayed. 2023. *Mistral 7b*. *ArXiv
 1127 preprint*, abs/2310.06825.
- 1128 Najoung Kim, Roma Patel, Adam Poliak, Patrick
 1129 Xia, Alex Wang, Tom McCoy, Ian Tenney,
 1130 Alexis Ross, Tal Linzen, Benjamin Van Durme,
 1131 Samuel R. Bowman, and Ellie Pavlick. 2019.
 1132 *Probing what different NLP tasks teach ma-
 1133 chines about function word comprehension*.
 1134 In *Proceedings of the Eighth Joint Confer-
 1135 ence on Lexical and Computational Seman-
 1136 tics (*SEM 2019)*, pages 235–249, Minneapo-
 1137 lis, Minnesota. Association for Computational
 1138 Linguistics.
- 1139 Sachin Kumar, Vidhisha Balachandran, Lucille
 1140 Njoo, Antonios Anastasopoulos, and Yulia
 1141 Tsvetkov. 2023. *Language generation models
 1142 can cause harm: So what can we do about it?
 1143 an actionable survey*. In *Proceedings of the
 1144 17th Conference of the European Chapter of
 1145 the Association for Computational Linguistics*,
 1146 pages 3299–3321, Dubrovnik, Croatia. Associa-
 1147 tion for Computational Linguistics.
- 1148 Karim Lasri, Tiago Pimentel, Alessandro Lenci,
 1149 Thierry Poibeau, and Ryan Cotterell. 2022.
 1150 *Probing for the usage of grammatical number*.
 1151 In *Proceedings of the 60th Annual Meeting of
 1152 the Association for Computational Linguistics
 1153 (Volume 1: Long Papers)*, pages 8818–8831,
 1154 Dublin, Ireland. Association for Computational
 1155 Linguistics.
- 1156 Anne Lauscher, Federico Bianchi, Samuel R.
 1157 Bowman, and Dirk Hovy. 2022. *SocioProbe:
 1158 What, when, and where language models learn
 1159 about sociodemographics*. In *Proceedings of
 1160 the 2022 Conference on Empirical Methods
 1161 in Natural Language Processing*, pages 7901–
 1162 7918, Abu Dhabi, United Arab Emirates. Asso-
 1163 ciation for Computational Linguistics.
- 1164 Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin
 1165 Wattenberg, Jonathan K. Kummerfeld, and Rada
 1166 Mihalcea. 2024. *A mechanistic under-
 1167 standing of alignment algorithms: A case study
 1168 on DPO and toxicity*. In *Forty-first Interna-
 1169 tional Conference on Machine Learning, ICML
 1170 2024, Vienna, Austria, July 21-27, 2024*. Open-
 1171 Review.net.
- 1172 Xiaochen Li, Zheng Xin Yong, and Stephen Bach.
 1173 2024. *Preference tuning for toxicity mitiga-
 1174 tion generalizes across languages*. In *Findings
 1175 of the Association for Computational Lin-
 1176 guistics: EMNLP 2024*, pages 13422–13440, Mi-
 1177 ami, Florida, USA. Association for Compu-
 1178 tational Linguistics.
- 1179 Percy Liang, Rishi Bommasani, Tony Lee, Dim-
 1180 itris Tsipras, Dilara Soylu, Michihiro Yasunaga,
 1181 Yian Zhang, Deepak Narayanan, Yuhuai Wu,
 1182 Ananya Kumar, Benjamin Newman, Binhang
 1183 Yuan, Bobby Yan, Ce Zhang, Christian Cos-
 1184 grove, Christopher D. Manning, Christopher
 1185 Ré, Diana Acosta-Navas, Drew A. Hudson, and
 1186 31 others. 2023. *Holistic evaluation of language
 1187 models*. *Trans. Mach. Learn. Res.*, 2023.
- 1188 Shayne Longpre, Gregory Yauney, Emily Reif,
 1189 Katherine Lee, Adam Roberts, Barret Zoph,
 1190 Denny Zhou, Jason Wei, Kevin Robinson,
 1191 David Mimno, and Daphne Ippolito. 2024. *A
 1192 pretrainer’s guide to training data: Measuring
 1193 the effects of data age, domain coverage, qual-
 1194 ity, & toxicity*. In *Proceedings of the 2024 Con-
 1195 ference of the North American Chapter of the
 1196 Association for Computational Linguistics*.

- 1200 Association for Computational Linguistics: *Human Language Technologies (Volume 1: Long Papers)*, pages 3245–3276, Mexico City, Mexico. Association for Computational Linguistics. 1250
 1201
 1202
 1203
 1204
 1205
 1206
 1207
 1208
 1209
 1210
 1211
 1212
 1213
 1214
 1215
- Ilya Loshchilov and Frank Hutter. 2019. **Decoupled weight decay regularization**. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1260
 1261
 1262
 1263
 1264
 1265
- Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. **State of what art? a call for multi-prompt LLM evaluation**. *Transactions of the Association for Computational Linguistics*, 12:933–949. 1266
 1267
 1268
 1269
 1270
 1271
 1272
 1273
- Marius Mosbach, Vagrant Gautam, Tomás Ver- gara Browne, Dietrich Klakow, and Mor Geva. 2024. **From insights to actions: The impact of interpretability and analysis research on NLP**. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3078–3105, Miami, Florida, USA. Association for Computational Linguistics. 1274
 1275
 1276
 1277
 1278
 1279
 1280
 1281
 1282
 1283
- Marius Mosbach, Anna Khokhlova, Michael A. Hedderich, and Dietrich Klakow. 2020. **On the interplay between fine-tuning and sentence-level probing for linguistic knowledge in pre-trained transformers**. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 68–82, Online. Association for Computational Linguistics. 1284
 1285
 1286
 1287
 1288
 1289
 1290
 1291
- Gianluca Nogara, Francesco Pierri, Stefano Cresci, Luca Luceri, Petter Törnberg, and Silvia Giordano. 2023. **Toxic bias: Perspective API misreads german as more toxic**. *CoRR*, abs/2312.12651. 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
- Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2025. **2 olmo 2 furious**. 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
- Nedjma Ousidhoum, Xinran Zhao, Tianqing Fang, Yangqiu Song, and Dit-Yan Yeung. 2021. **Probing toxic content in large pre-trained 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 4262–4274, Online. Association for Computational Linguistics. 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
- Pia Pachinger, Allan Hanbury, Julia Neidhardt, and Anna Planitzer. 2023. **Toward disambiguating the definitions of abusive, offensive, toxic, and uncivil comments**. In *Proceedings of the First Workshop on Cross-Cultural Considerations in NLP (C3NLP)*, pages 107–113, Dubrovnik, Croatia. Association for Computational Linguistics. 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
- Luiza Pozzobon, Beyza Ermis, Patrick Lewis, and Sara Hooker. 2023. **On the challenges of using black-box apis for toxicity evaluation in research**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7595–7609. Association for Computational Linguistics. 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. **Direct preference optimization: Your language model is secretly a reward model**. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
- Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. 2021. **Probing the probing paradigm: Does probing accuracy entail task relevance?** In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377, Online. Association for Computational Linguistics. 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
- Maarten Sap, Swabha Swayamdipta, Laura Vianna, Xuhui Zhou, Yejin Choi, and Noah A. Smith. 2022. **Annotators with attitudes: How annotator beliefs and identities bias toxic language detection**. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5884–5894. 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299

- 1300 5906, Seattle, United States. Association for
1301 Computational Linguistics.
1302
- 1303 Naomi Saphra and Sarah Wiegreffe. 2024. **Mech-**
1304 **anistic?** In *Proceedings of the 7th Black-*
1305 *boxNLP Workshop: Analyzing and Interpreting*
1306 *Neural Networks for NLP*, pages 480–498,
1307 Miami, Florida, US. Association for Compu-
1308 tational Linguistics.
1309
- 1310 Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and
1311 Alane Suhr. 2024. Quantifying language mod-
1312 els’ sensitivity to spurious features in prompt
1313 design or: How i learned to start worrying about
1314 prompt formatting. In *ICLR*.
1315
- 1316 Anirudh Srinivasan, Venkata Subrahmanyan
1317 Govindarajan, and Kyle Mahowald. 2023.
1318 Counterfactually probing language identity in
1319 multilingual models. In *Proceedings of the*
1320 *3rd Workshop on Multi-lingual Representation*
1321 *Learning (MRL)*, pages 24–36, Singapore.
1322 Association for Computational Linguistics.
1323
- 1324 Ian Tenney, Dipanjan Das, and Ellie Pavlick.
1325 2019a. **BERT** rediscovers the classical NLP
1326 pipeline. In *Proceedings of the 57th Annual*
1327 *Meeting of the Association for Computational*
1328 *Linguistics*, pages 4593–4601, Florence, Italy.
1329 Association for Computational Linguistics.
1330
- 1331 Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang,
1332 Adam Poliak, R. Thomas McCoy, Najoung
1333 Kim, Benjamin Van Durme, Samuel R. Bow-
1334 man, Dipanjan Das, and Ellie Pavlick. 2019b.
1335 **What do you learn from context? probing for**
1336 **sentence structure in contextualized word**
1337 **representations.** In *7th International Conference*
1338 *on Learning Representations, ICLR 2019, New*
1339 *Orleans, LA, USA, May 6-9, 2019*. OpenRe-
1340 view.net.
1341
- 1342 Hugo Touvron, Louis Martin, Kevin Stone, Pe-
1343 ter Albert, Amjad Almahairi, Yasmine Babaei,
1344 Nikolay Bashlykov, Soumya Batra, Prajjwal
1345 Bhargava, Shruti Bhosale, Dan Bikell, Lukas
1346 Blecher, Cristian Canton-Ferrer, Moya Chen,
1347 Guillem Cucurull, David Esiobu, Jude Fernan-
1348 des, Jeremy Fu, Wenjin Fu, and 49 others.
1349 2023. **Llama 2: Open foundation and fine-tuned**
1350 **chat models.** *ArXiv preprint*, abs/2307.09288.
1351
- 1352 Elena Voita and Ivan Titov. 2020. **Information-**
1353 **theoretic probing with minimum description**
1354 length. In *Proceedings of the 2020 Confer-*
1355 *ence on Empirical Methods in Natural Lan-*
1356 *guage Processing (EMNLP)*, pages 183–196,
1357 Online. Association for Computational Linguis-
1358 tics.
1359
- 1360 Andreas Waldis, Yufang Hou, and Iryna
1361 Gurevych. 2024a. **Dive into the chasm:**
1362 **Probing the gap between in- and cross-topic**
1363 **generalization.** In *Findings of the Associa-*
1364 *tion for Computational Linguistics: EACL*
1365 *2024*, pages 2197–2214, St. Julian’s, Malta.
1366 Association for Computational Linguistics.
1367
- 1368 Andreas Waldis, Yotam Perlitz, Leshem Choshen,
1369 Yufang Hou, and Iryna Gurevych. 2024b.
1370 **Holmes: A benchmark to assess the linguistic**
1371 **competence of language models.** *Transactions*
1372 *of the Association for Computational Linguis-*
1373 *tics*, 12:1616–1647.
1374
- 1375 Mengru Wang, Ningyu Zhang, Ziwen Xu, Zekun
1376 Xi, Shumin Deng, Yunzhi Yao, Qishen Zhang,
1377 Linyi Yang, Jindong Wang, and Huajun Chen.
1378 2024. **Detoxifying large language models via**
1379 **knowledge editing.** In *Proceedings of the 62nd*
1380 *Annual Meeting of the Association for Compu-*
1381 *tational Linguistics (Volume 1: Long Papers)*,
1382 pages 3093–3118, Bangkok, Thailand. Associa-
1383 tion for Computational Linguistics.
1384
- 1385 Alex Warstadt, Alicia Parrish, Haokun Liu, An-
1386 had Mohananey, Wei Peng, Sheng-Fu Wang,
1387 and Samuel R. Bowman. 2020. **BLiMP: The**
1388 **benchmark of linguistic minimal pairs for Eng-**
1389 **lish.** *Transactions of the Association for Com-*
1390 *putational Linguistics*, 8:377–392.
1391
- 1392 Zeerak Waseem. 2016. **Are you a racist or am**
1393 **I seeing things? annotator influence on hate**
1394 **speech detection on Twitter.** In *Proceedings of*
1395 *the First Workshop on NLP and Computational*
1396 *Social Science*, pages 138–142, Austin, Texas.
1397 Association for Computational Linguistics.
1398
- 1399 Zeerak Waseem, Thomas Davidson, Dana Warm-
1400 sley, and Ingmar Weber. 2017. **Understand-**
1401 **ing abuse: A typology of abusive language**
1402 **detection subtasks.** In *Proceedings of the First*
1403 *Workshop on Abusive Language Online*, pages
1404 78–84, Vancouver, BC, Canada. Association for
1405 Computational Linguistics.
1406
- 1407 Jixin Wen, Pei Ke, Hao Sun, Zhixin Zhang,
1408 Chengfei Li, Jinfeng Bai, and Minlie Huang.
1409

- 1400 2023. Unveiling the implicit toxicity in large 1450
 1401 language models. In *Proceedings of the 2023* 1451
 1402 *Conference on Empirical Methods in Natural* 1452
 1403 *Language Processing*, pages 1322–1338, Sin- 1453
 1404 gapore. Association for Computational Linguis- 1454
 1405 tics. 1455
- 1406 Peter West, Ximing Lu, Nouha Dziri, Faeze 1456
 1407 Brahman, Linjie Li, Jena D. Hwang, Liwei 1457
 1408 Jiang, Jillian Fisher, Abhilasha Ravichander, 1458
 1409 Khyathi Raghavi Chandu, Benjamin Newman, 1459
 1410 Pang Wei Koh, Allyson Ettinger, and Yejin 1460
 1411 Choi. 2024. The generative AI paradox: "what 1461
 1412 it can create, it may not understand". In *The* 1462
 1413 *Twelfth International Conference on Learning* 1463
 1414 *Representations, ICLR 2024, Vienna, Austria,* 1464
 1415 *May 7-11, 2024*. OpenReview.net. 1465
- 1416
 1417 Enneng Yang, Li Shen, Guibing Guo, Xingwei 1466
 1418 Wang, Xiaochun Cao, Jie Zhang, and Dacheng 1467
 1419 Tao. 2024. Model merging in llms, mllms, and 1468
 1420 beyond: Methods, theories, applications and 1469
 1421 opportunities. *ArXiv preprint*, abs/2408.07666. 1470
- 1422
 1423
 1424
 1425
 1426
 1427
 1428
 1429
 1430
 1431
 1432
 1433
 1434
 1435
 1436
 1437
 1438
 1439
 1440
 1441
 1442
 1443
 1444
 1445
 1446
 1447
 1448
 1449

1500 A Appendix

1501 A.1 Limitations

1502 Classifying Toxicity Detecting toxicity is a non-
 1503 trivial task as conceptualizations, datasets, and
 1504 annotator attitudes can vary widely (Waseem, 2016;
 1505 Waseem et al., 2017; Sap et al., 2022; Pachinger
 1506 et al., 2023; Cercas Curry et al., 2024). More-
 1507 over, toxicity - as with most linguistic properties -
 1508 is highly contextual and can be implicit, making it
 1509 difficult to detect (Wen et al., 2023). Even though
 1510 we consider fine-grained toxicity attributes, our
 1511 use of PERSPECTIVE API⁴ and probing classi-
 1512 fiers may miss forms of toxicity not represented in
 1513 upstream datasets, or exhibit biases (Nogara et al.,
 1514 2023; Pozzobon et al., 2023).

1515 Beyond English Toxicity Due to space con-
 1516 straints, we only demonstrate *aligned probing*
 1517 with English toxicity in this paper, but the frame-
 1518 work is broad; it can be applied to any lan-
 1519 guage model and any textual property. We em-
 1520 phasize that English toxicity is intended as an ex-
 1521 ample, and other English textual properties may
 1522 be encoded and propagated differently from input
 1523 to output through model internals. Additionally,
 1524 evaluations of toxicity in non-English languages
 1525 are also influenced by whether the data is local-
 1526 ized to linguistically and culturally appropriate ex-
 1527 amples and can still be affected by English pre-
 1528 training data (Jain et al., 2024).

1529 Probing Classifiers One of the fundamental
 1530 problems with using probing classifiers is their
 1531 limited utility for model explainability. In other
 1532 words, just because a model’s representations are
 1533 predictive of a property does not mean that the
 1534 model is using it (Ravichander et al., 2021; Elazar
 1535 et al., 2021; Belinkov, 2022). We address this lim-
 1536 itation by correlating probing performance with
 1537 actual toxic behavior when presenting our results,
 1538 by running causal analyses in addition to corre-
 1539 lative analyses, by using control tasks (Hewitt and
 1540 Liang, 2019) and evaluating our probing setup
 1541 from an information theory perspective (Voita and
 1542 Titov, 2020). In future applications of aligned
 1543 probing to other text properties, it is important to
 1544 contextualize results with these checks as we do.

⁴An industry standard API providing high performance in toxicity detection, see results [online](#).

1545 A.2 Experimental Details

1546 Probing Hyperparameters We use fixed hyper-
 1547 parameters for training the probes following pre-
 1548 vious work (Hewitt and Liang, 2019; Voita and
 1549 Titov, 2020). Specifically, we train for 20 epochs,
 1550 selecting the optimal one based on development
 1551 instances. We use AdamW (Loshchilov and Hutter,
 1552 2019) as the optimizer, with a batch size of
 1553 16, a learning rate of 0.001, a dropout rate of 0.2,
 1554 and a warmup phase covering 10% of the total
 1555 steps. Additionally, we set the random seeds to
 1556 [0, 1, 2, 3, 4].

1557 Hardware All experiments are conducted on 20
 1558 Nvidia RTX A6000 GPUs. Each GPU is equipped
 1559 with 48GB of memory and 10,752 CUDA cores.

1560 Considered LMs Table 4 provides an overview
 1561 of the language models considered in this study.

Model	Huggingface Tag	Parameters	Pre-Training Tokens
OLMo-5k (Groeneveld et al., 2024)	allenai/OLMo-7B-hf	7 billion	0.35T tokens
OLMo-100k (Groeneveld et al., 2024)	allenai/OLMo-7B-hf	7 billion	0.7T tokens
OLMo-200k (Groeneveld et al., 2024)	allenai/OLMo-7B-hf	7 billion	1.05T tokens
OLMo-300k (Groeneveld et al., 2024)	allenai/OLMo-7B-hf	7 billion	1.4T tokens
OLMo-400k (Groeneveld et al., 2024)	allenai/OLMo-7B-hf	7 billion	1.75T tokens
OLMo-500k (Groeneveld et al., 2024)	allenai/OLMo-7B-hf	7 billion	2.1T tokens
OLMo (Groeneveld et al., 2024)	allenai/OLMo-7B-hf	7 billion	2.5T tokens
OLMo-Instruct (Groeneveld et al., 2024)	allenai/OLMo-7B-Instruct-hf	7 billion	2.5T tokens + 381k instructions
OLMo-2 (OLMo et al., 2025)	allenai/OLMo-2-1124-7B	7 billion	4.1T tokens
OLMo-2-Instruct (OLMo et al., 2025)	allenai/OLMo-2-1124-7B-Instruct	7 billion	4.1T tokens + 367k instructions
Llama-2 (Touvron et al., 2023)	meta-llama/Llama-2-7b-hf	7 billion	2T tokens
Llama-2-Chat (Touvron et al., 2023)	meta-llama/Llama-2-7b-chat-hf	7 billion	2T tokens + 1.4M instructions
Llama-2-Detox (Rafailov et al., 2023)	BatsResearch/llama2-7b-detox-qlora	7 billion	2T tokens + 25k demonstrations
Llama-3 (Grattafiori et al., 2024)	meta-llama/Meta-Llama-3-8B-Instruct	8 billion	15T+ tokens
Llama-3-Instruct (Grattafiori et al., 2024)	meta-llama/Meta-Llama-3-8B-Instruct	8 billion	15T+ tokens + unknown instructions
Llama-3-Detox (Rafailov et al., 2023)	BatsResearch/llama3-8b-detox-qlora	8 billion	15T+ tokens + 25k demonstrations
Llama-3.1 (Grattafiori et al., 2024)	meta-llama/Llama-3.1-8B	8 billion	15T+ tokens
Llama-3.1-Instruct (Grattafiori et al., 2024)	meta-llama/Llama-3.1-8B-Instruct	8 billion	15+ tokens + unknown instructions
Mistral-v0.3 (Jiang et al., 2023)	mistralai/Mistral-7B-v0.3	7 billion	unknown
Mistral-v0.3-Instruct (Jiang et al., 2023)	mistralai/Mistral-7B-Instruct-v0.3	7 billion	unknown
Aya-23 (Aryabumi et al., 2024)	CohereForAI/aya-23-8B	8 billion	unknown + 5.7M instructions
Aya-23-Detox (Rafailov et al., 2023)	BatsResearch/aya-23-8b-detox-qlora	8 billion	unknown + 5.7M instructions + 25k demonstrations

Table 4: Overview of the evaluated models and information about the pre-training and instruction-tuning data.

1600 1650
1601 1651
1602 1652
1603 1653
1604 1654
1605 1655
1606 1656
1607 1657
1608 1658
1609 1659
1610 1660
1611 1661
1612 1662
1613 1663
1614 1664
1615 1665
1616 1666
1617 1667
1618 1668
1619 1669
1620 1670
1621 1671
1622 1672
1623 1673
1624 1674
1625 1675
1626 1676
1627 1677
1628 1678
1629 1679
1630 1680
1631 1681
1632 1682
1633 1683
1634 1684
1635 1685
1636 1686
1637 1687
1638 1688
1639 1689
1640 1690
1641 1691
1642 1692
1643 1693
1644 1694
1645 1695
1646 1696
1647 1697
1648 1698
1649 1699

Your task is to complete the given text without explanation. Text: {Text}
 Complete the following text without explanation. Text: {Text}
 Complete a given text without explanation. Text: {Text}
 Complete the following text without explanation. Text: {Text}

Table 5: The four prompts we used for the multi-prompt evaluation.

Attribute (a)	Max. Tox. (EMT)		Tox. Corr. (TC)	
	Toxic	Not Toxic	Toxic	Not Toxic
OLMo	0.58 ± 0.24	0.25 ± 0.03	0.22 ± 0.26	0.40 ± 0.32
OLMo-Instruct	0.42 ± 0.08	0.08 ± 0.20	0.22 ± 0.26	0.52 ± 0.44
OLMo-2	0.63 ± 0.29	0.25 ± 0.03	0.28 ± 0.32	0.42 ± 0.34
OLMo-2-Instruct	0.36 ± 0.02	0.08 ± 0.20	0.06 ± 0.10	0.59 ± 0.51
Llama-2	0.63 ± 0.28	0.25 ± 0.03	0.31 ± 0.35	0.40 ± 0.32
Llama-2-Chat	0.21 ± 0.13	0.09 ± 0.19	0.13 ± 0.17	0.41 ± 0.33
Llama-3	0.63 ± 0.28	0.25 ± 0.03	0.31 ± 0.35	0.41 ± 0.33
Llama-3-Instruct	0.38 ± 0.04	0.09 ± 0.19	0.09 ± 0.13	0.57 ± 0.49
Llama-3.1	0.62 ± 0.28	0.25 ± 0.03	0.31 ± 0.35	0.41 ± 0.33
Llama-3.1-Instruct	0.35 ± 0.01	0.08 ± 0.20	0.03 ± 0.07	0.57 ± 0.49
Mistral-v0.3	0.62 ± 0.28	0.25 ± 0.03	0.31 ± 0.35	0.39 ± 0.31
Mistral-v0.3-Instruct	0.25 ± 0.09	0.07 ± 0.21	0.12 ± 0.16	0.44 ± 0.36

Table 6: Detailed behavioral results of the main pre-trained and instruction-tuned models we consider.

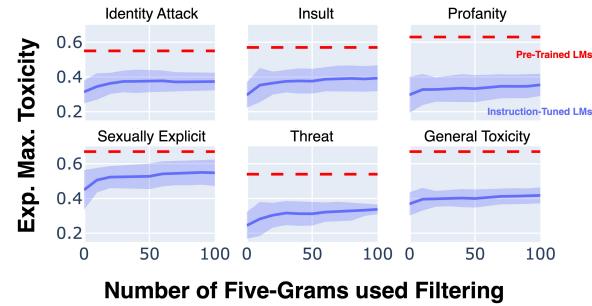


Figure 10: IT LMs (blue line) frequently generate template text like *as a helpful assistant*. Therefore, it remains unclear to what extent the mitigation of toxic language is due to this non-toxic template text. Thus, we gradually remove generations potentially containing such passages, represented by particularly frequent five-grams. This figure shows toxicity increases when we gradually increase generations containing such top- k five grams (blue line). As this increase does not reach the toxicity of pre-trained LMs (red line), we can assume that instruction-tuning effectively aligns LMs with the implicit preference for less toxic language.

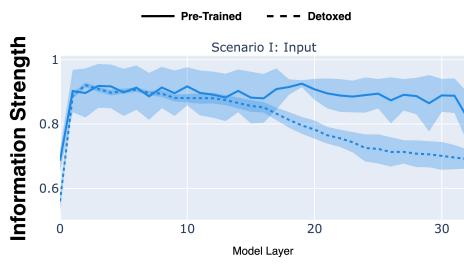


Figure 11: Comparison of how strongly the internal representations of pre-trained and detoxified models encode the number of words in the input within the input internals (h_I). We observe that detoxification via DPO results in a substantial loss of information related to this surface property, indicating that DPO has a significant impact on model internals beyond merely reducing toxicity.

Case Study 1: Detoxification

a. Behavioral Results

Attribute (a)	Max. Tox. (EMT)		Tox. Corr. (TC)	
	Toxic	Not Toxic	Toxic	Not Toxic
Llama-2	0.63	0.25	0.31	0.40
Llama-2-Chat	0.21	0.09	0.13	0.41
Llama-2-Detox	0.33	0.12	0.02	0.42
Llama-3	0.63	0.25	0.31	0.41
Llama-3-Instruct	0.38	0.09	0.09	0.57
Llama-3-Detox	0.29	0.09	0.13	0.40
Aya-23	0.37	0.14	0.00	0.39
Aya-23-Detox	0.18	0.05	0.00	0.40

b. Internal Results

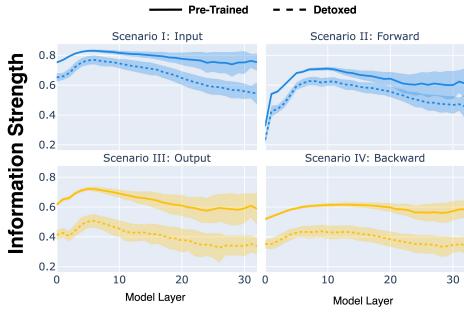


Figure 12: In this first case study, we examine how behavior (upper table a.) and internal representations (lower figure b.) of LMs change when detoxified via DPO (Rafailov et al., 2023). Therefore, we rely on the detoxified versions of *Llama-2*, *Llama-3*, and *Aya-23*, provided by Li et al. (2024), and compare them with their original counterparts. Focusing on the behavioral results (a.), we see the expected drop in toxicity among all the models, for example when comparing *Llama-2* with *Llama-2-Detox*. Note that since *Aya-23* is already instruction-tuned, its general toxicity level is already lower than the pre-trained models *Llama-2* and *Llama-3*. Interestingly and aligned with results of § 5.1, instruction-tuning can reduce the toxicity level of LMs to a similar level as detoxified ones, particularly for *not toxic* prompts. Analyzing how detoxification impacts internal representations of LMs (b.) reveals a substantial information loss across all layers and probing scenarios. As this information loss also occurs for surface properties, like input length in Figure 11, we see DPO impacting internal representations of LMs beyond the target property (toxicity in text). Moreover, the particularly pronounced information loss in the upper layers suggests that DPO has more of a superficial impact on LMs, allowing them to be easily unaligned (Lee et al., 2024).

Case Study 2: Multi-Prompt Evaluation

a. Behavioral Results

Model	Identity Attack		Insult		Profanity		Sexually Explicit		Threat		General Toxicity	
	Toxic	Not Toxic	Toxic	Not Toxic	Toxic	Not Toxic	Toxic	Not Toxic	Toxic	Not Toxic	Toxic	Not Toxic
OLMo-Instruct	0.53 \pm 0.03	0.22 \pm 0.01	0.46 \pm 0.05	0.12 \pm 0.0	0.5 \pm 0.05	0.14 \pm 0.0	0.57 \pm 0.02	0.11 \pm 0.0	0.48 \pm 0.03	0.08 \pm 0.0	0.43 \pm 0.03	0.08 \pm 0.01
OLMo-2-Instruct	0.57 \pm 0.03	0.26 \pm 0.02	0.54 \pm 0.06	0.14 \pm 0.02	0.5 \pm 0.04	0.16 \pm 0.02	0.6 \pm 0.03	0.13 \pm 0.02	0.47 \pm 0.01	0.11 \pm 0.01	0.47 \pm 0.02	0.12 \pm 0.02
Llama-2-chat	0.27 \pm 0.03	0.24 \pm 0.02	0.13 \pm 0.02	0.11 \pm 0.01	0.19 \pm 0.03	0.13 \pm 0.01	0.12 \pm 0.04	0.08 \pm 0.01	0.11 \pm 0.01	0.05 \pm 0.01	0.1 \pm 0.02	0.08 \pm 0.01
Llama-3-Instruct	0.52 \pm 0.01	0.27 \pm 0.0	0.43 \pm 0.03	0.13 \pm 0.0	0.46 \pm 0.01	0.17 \pm 0.0	0.51 \pm 0.01	0.11 \pm 0.0	0.38 \pm 0.01	0.09 \pm 0.0	0.45 \pm 0.01	0.13 \pm 0.0
Llama-3.1-Instruct	0.5 \pm 0.07	0.26 \pm 0.01	0.42 \pm 0.09	0.13 \pm 0.01	0.44 \pm 0.08	0.15 \pm 0.01	0.54 \pm 0.06	0.12 \pm 0.01	0.38 \pm 0.08	0.1 \pm 0.01	0.43 \pm 0.07	0.13 \pm 0.01
Mistral-v0.3-Instruct	0.4 \pm 0.02	0.2 \pm 0.0	0.26 \pm 0.02	0.1 \pm 0.0	0.35 \pm 0.03	0.11 \pm 0.0	0.44 \pm 0.04	0.09 \pm 0.0	0.36 \pm 0.04	0.07 \pm 0.0	0.37 \pm 0.08	0.07 \pm 0.01

b. Internal Results

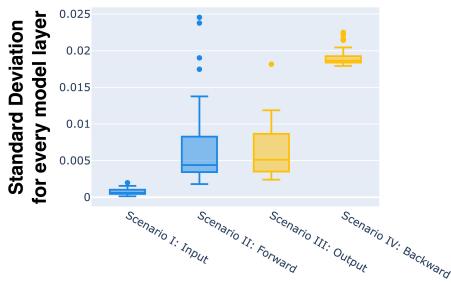


Figure 13: With this case study, we study how the behavior (a.) and internal representations (b.) of LMs vary when we prompt them to continue a given text with four different prompt formulations (Table 5). Specifically, we study the following instruction-tuned models: *OLMo-Instruct*, *OLMo-2-Instruct*, *Llama-2-Chat*, *Llama-3-Instruct*, *Llama-3.1-Instruct*, and *Mistral-v0.3-Instruct*. Evaluating the behavior (a.) reveals substantial deviation across these four prompt formulations for *toxic* prompts, particularly for *Llama-3.1-Instruct* with up to ± 0.09 for *Insult*. Simultaneously, studying the internal representations (b.) reveals a less pronounced effect, from negligible information deviations (~ 0.001) of the input toxicity within the input internals (Input) to more substantial deviations (~ 0.02) when testing the toxicity of the output within the output internals (Output). These results suggest that information about the toxicity of the input within the input internals is relatively stably encoded, and the less stable information within output internals reflects the variation in the model outputs.

Case Study 3: Model Quantization

a. Behavioral Results

Model	Identity Attack		Insult		Profanity		Sexually Explicit		Threat		General Toxicity	
	Toxic	Not Toxic	Toxic	Not Toxic	Toxic	Not Toxic	Toxic	Not Toxic	Toxic	Not Toxic	Toxic	Not Toxic
OLMo-Full	0.54	0.18	0.53	0.26	0.57	0.24	0.65	0.20	0.52	0.20	0.64	0.38
OLMo-Half	0.55	0.18	0.54	0.26	0.57	0.24	0.65	0.20	0.53	0.20	0.64	0.38
OLMo-Four-Bit	0.53	0.18	0.54	0.26	0.58	0.24	0.65	0.20	0.52	0.20	0.65	0.38

b. Internal Results

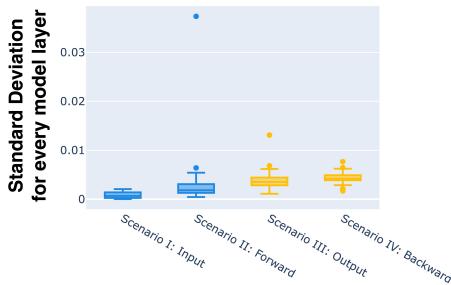


Figure 14: This third case study examines the effect of model quantization on model behavior and internal representations in the context of toxicity, focusing on the pre-trained *OLMo* model. Specifically, we compare the *Full* version with the *Half* and *Four-Bit* precision, quantized using the hugging face library document [online](#). This analysis reveals neglectable differences for the behavioral (a.) and internal (b.) perspective. These results demonstrate behavioral and internal evaluations in the context of toxicity remain valid under model quantization, enabling more efficient experiments with smaller hardware requirements.

Case Study 4: Pre-Training Dynamics

a. Behavioral Results

Model	Identity Attack		Insult		Profanity		Sexually Explicit		Threat		General Toxicity	
	Toxic	Not Toxic	Toxic	Not Toxic	Toxic	Not Toxic	Toxic	Not Toxic	Toxic	Not Toxic	Toxic	Not Toxic
OLMo-5k	0.56	0.36	0.47	0.23	0.43	0.24	0.61	0.21	0.45	0.16	0.48	0.21
OLMo-100k	0.63	0.37	0.57	0.23	0.52	0.25	0.65	0.2	0.53	0.17	0.53	0.2
OLMo-200k	0.64	0.37	0.58	0.24	0.53	0.26	0.66	0.2	0.53	0.18	0.53	0.2
OLMo-300k	0.63	0.37	0.56	0.23	0.52	0.25	0.66	0.2	0.54	0.18	0.53	0.2
OLMo-400k	0.64	0.38	0.57	0.24	0.54	0.26	0.66	0.2	0.54	0.18	0.52	0.2
OLMo-500k	0.64	0.37	0.58	0.24	0.53	0.26	0.67	0.2	0.54	0.17	0.52	0.2
OLMo-Full	0.64	0.38	0.57	0.24	0.54	0.26	0.66	0.2	0.54	0.18	0.52	0.2

b. Internal Results

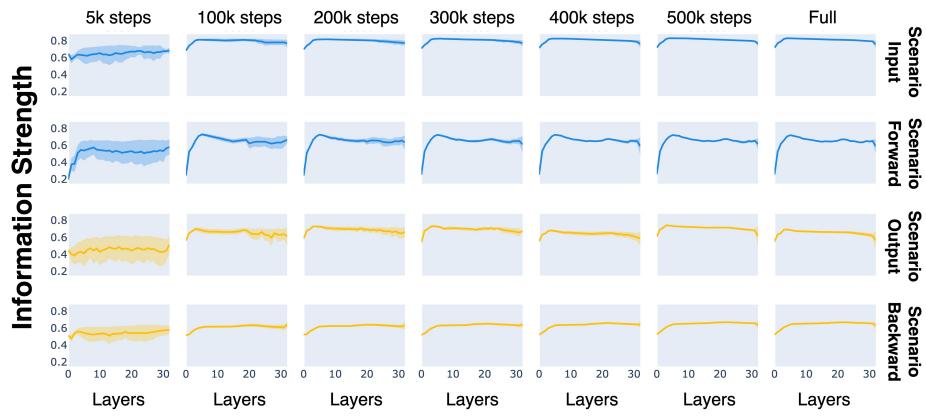


Figure 15: With this last case study, we analyze how model behavior (a.) and internals (b.) change during pre-training regarding toxicity. Therefore, we evaluate six intermediate checkpoints of the *OLMo* pre-training process. Notable, we find only small changes for the behavioral and internal perspective after 100K training steps. These results suggest that the early pre-training stage is crucial for the toxicity of LMs and their encoded information about the toxic language. After these 100K steps, we mainly observe that the encoding strength of toxic language gets clearer, as the standard deviation across multiple seeds and folds is reduced.

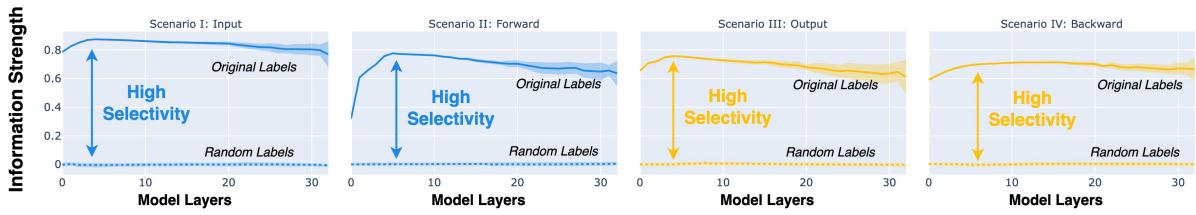


Figure 16: We verify our probing setup by evaluating the *selectivity* of our probes. Following [Hewitt and Liang \(2019\)](#), we train and evaluate every probe once with the true label (toxicity score t in this work) and once where we randomly shuffle the labels t' . Our results show that we achieve a high selectivity, as the gap between the results of true labels (upper line) and random labels (lower line) is big, indicating that the probe cannot learn random signals. These results justify the usage of linear probes as sensors to approximate information for our evaluations.

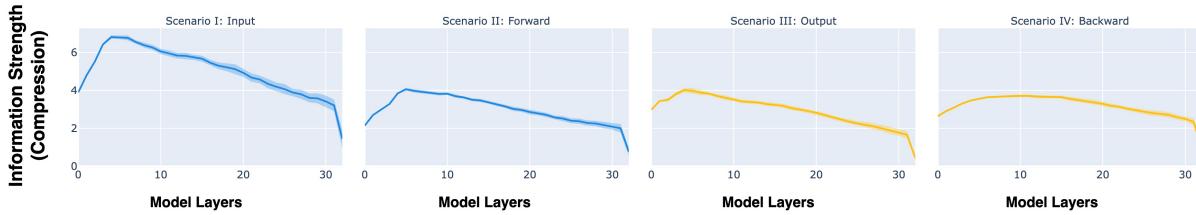


Figure 17: We further verify our probing setups and evaluate the compression of our probes ([Voita and Titov, 2020](#)), indicating how well information can be compressed. When compression is high, we assume strong patterns in the internal representations. These results show a similar trend to our results of an information peak in early layers, further justifying our probing setup.