Ablation is Not Enough to Emulate DPO: How Neuron Dynamics Drive Toxicity Reduction

Yushi Yang[*]
University of Oxford

Filip Sondej
Independent

Harry Mayne
University of Oxford

Adam Mahdi
University of Oxford

Abstract

Safety fine-tuning algorithms are commonly used to fine-tune language models to reduce harmful outputs, but the exact internal mechanisms of how those models achieve this remain unclear. In studying direct preference optimisation (DPO) for toxicity reduction, current explanations claim that DPO works by dampening the most toxic MLP neurons to learn an offset to avert toxic regions in the residual stream. However, by ablating the most toxic neurons and applying activation patching, we find this explanation incomplete. By projecting neuron activation changes onto a toxicity probe, we find that only 31.8% of toxicity reduction comes from dampened toxic neurons. Instead, DPO reduces toxicity by accumulating effects across multiple neuron groups, both reducing writing in the toxic direction and promoting anti-toxicity in the residual stream. Moreover, DPO gives noisy adjustments to neuron activations, with many neurons actually increasing toxicity. This indicates that DPO is a balancing process between opposing neuron effects to achieve toxicity reduction. [2]

1 Introduction

The generality of an LLM's capabilities means the model also learns to encode undesirable behaviours, such as producing toxic, biased, or hallucinated outputs [6, 5, 19]. To address these issues, researchers have developed safety fine-tuning algorithms, such as proximal policy optimization (PPO) [15] and direct preference optimization (DPO) [14], to reduce undesirable outputs.

Recent studies showed that these safety fine-tuning algorithms cause minimal changes to the parameters of pre-trained models, and the undesirable behaviours are hidden rather than fully eliminated [11, 8, 9]. However, the exact mechanisms through which small parameter changes lead to the suppression of undesirable behaviours remain unclear. One explanation proposed when studying the DPO algorithm for toxicity reduction, claimed that DPO reduces toxicity by dampening the activations of the most toxic MLP neurons, creating an offset to avert toxic regions in the residual stream [11]. Our study tests this claim by tracking the writing of the toxic feature direction detected by a probe across MLP layers and neurons in GPT-2 medium. Specifically, we project neuron activation changes onto the toxicity probe direction to quantify per-neuron toxicity adjustments, providing a precise mechanistic understanding of DPO's mechanisms. Our findings are:

- *DPO does more than dampening toxic neurons.* By ablating the most toxic neurons and activation patching on the pre-trained model, we find that toxicity levels remain higher

[*]Correspondence: Yushi Yang, <yushi.yang@oii.ox.ac.uk>

[2]The code is available at: https://github.com/Yushi-Y/dpo-toxic-neurons.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).
Workshop on Foundation Model Interventions (MINT).

Table 1: **Toxicity, Perplexity (PPL) and F1 scores after ablating and patching most toxic neurons**. Ablating the most toxic neurons or patching their activations to post-DPO levels results in some toxicity reduction, but this effect remains limited compared to DPO's impact.

Model	Intervention	Toxicity	PPL	F1
GPT2	None	0.453	21.70	0.193
GPT2	Ablate top 100 toxic neurons	0.403	21.99	0.192
GPT2	Ablate top 200 toxic neurons	0.405	22.41	0.192
GPT2	Ablate top 1000 toxic neurons	0.436	27.34	0.184
GPT2	Ablate top 100 positively activated toxic neurons	0.384	21.78	0.193
GPT2	Ablate top 200 positively activated toxic neurons	0.366	21.83	0.193
GPT2	Ablate top 1000 positively activated toxic neurons	0.320	30.04	0.191
GPT2	Ablate top 2000 positively activated toxic neurons	0.319	29.07	0.189
GPT2	Patch all dampened toxic neurons to post-DPO levels	0.335	21.69	0.190
DPO	None	0.208	23.34	0.195
DPO	Scale the key vectors on top 7 toxic neurons (x2)	0.487	21.72	0.192
DPO	Scale the key vectors on top 7 toxic neurons (x5)	0.555	23.36	0.188
DPO	Scale the key vectors on top 7 toxic neurons (x10)	0.458	37.33	0.183

than when DPO is applied, indicating that dampened toxic neurons alone [11] do not fully account for DPO's effect.

- *A significant part of DPO's effect comes from actively writing anti-toxicity into the residual stream.* By projecting onto the toxicity probe, our analysis shows that dampened toxic neurons only account for 31.8% of the total toxicity reduction. DPO not only writes less in the toxic direction but also promotes anti-toxicity by activating more on anti-toxic neurons, or pushing inactive toxic neurons' activations further below zero.
- *Many neurons modified by DPO actually increase toxicity.* DPO introduces noisy activation adjustments across neurons, with roughly half writing less in the toxic direction and the other half writing more, creating a trade-off. This suggests that DPO balances opposing neuron effects to achieve overall toxicity reduction.

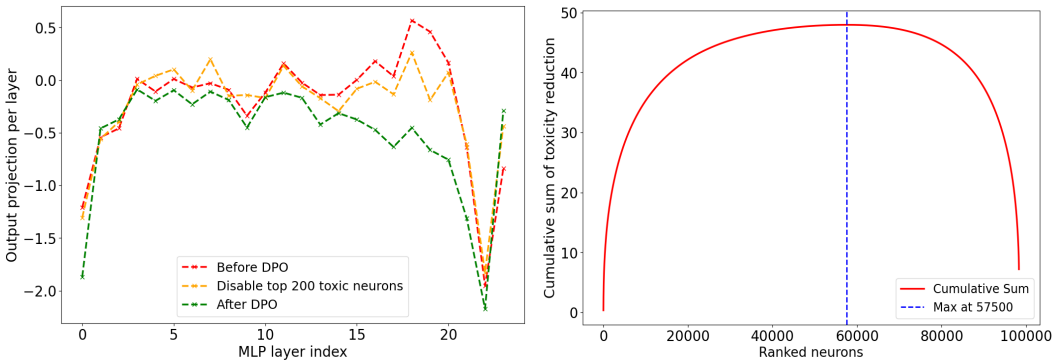
2 Background: Mechanisms of fine-tuning algorithms

Several studies have theorised how fine-tuning algorithms alter the capabilities of pre-trained models. Jain et al. [8] fine-tuned a language model on synthetic tasks and showed that the model develops "wrappers" in its later layers — small, localised adjustments to its pre-training abilities to optimise for each task. In a similar setting, Jain et al. [9] found that safety fine-tuning methods work by minimally transforming MLP weights to project unsafe inputs into its weights' null space. Wei et al. [18] demonstrated the brittleness of safety fine-tuning methods, showing that pruning just 3% of targeted model parameters can unlock the model from aligned behaviours.

In our reference study, Lee et al. [11] examined how the DPO algorithm works internally to reduce toxicity. Referring to the first and second weight vectors for an MLP neuron as the *key vector* and *value vector*, respectively [7] (see Appendix A for full notations), Lee et al. [11] proposed that DPO primarily reduces toxicity by suppressing the most toxic MLP neurons, whose value vectors align the most with a toxicity linear probe, thus shifting the model activation out of toxic regions associated with these value vectors. Our study tests this claim and finds it incomplete, as discussed further.

3 Experimental setup

To test Lee et al. [11]'s claims, we replicate their experimental setup, including the same language models, toxicity-eliciting prompts, probe extraction, and evaluation metrics. Specifically, we focus on GPT-2 medium with 355M parameters, 24 layers, a residual stream dimension of 1024, and an MLP hidden layer dimension of 4096 [11]. We also use the DPO-ed version of GPT2-medium [11] fine-tuned on 24,576 pairs of toxicity data generated by PPLM pipeline [3] on Wikitext-2 prompts.



(a) MLP layer outputs projected to toxicity probe. (b) Cumulative toxicity reduction ranked by neurons.

Figure 1: **Toxicity projection to the toxic probe across MLP layers and neurons.** (a) Output projections of MLP layers before DPO (red), after ablating top 200 toxic neurons (yellow), and after DPO (green). (b) The cumulative sum of toxicity reduction contributed by neurons, with neurons ranked from highest to lowest toxicity reduction.

To elicit toxic outputs, we use the "challenge" subset of REALTOXICITYPROMPTS [6], which contains 1,199 highly toxic prompts. We use the same linear toxic probe vector v_{toxic} in [11] to capture the aggregated toxicity feature direction in GPT2-medium. The toxicity probe was trained on a binary classification task using the Jigsaw toxic comment classification dataset (561,808 comments) [2] on the last layer of the residual stream in GPT2-medium [11]. We identify the most toxic neurons as those whose value vectors have the highest cosine similarity to v_{toxic}, which also clearly projects into toxic tokens in the vocabulary space [11]. To evaluate both toxicity and language quality in generated text, we measure *toxicity scores* via the Perspective API [7], *perplexity* on Wikitext-2 dataset [13], and *F1 scores* by matching the tokens in 2,000 Wikipedia sentences [4].

4 Tracking toxic feature reduction across neurons

4.1 Ablating toxic neurons

To test Lee et al. [11]'s claim that DPO dampens most toxic neurons to avert associated toxic regions, we ablate the activations of up to 2,000 toxic neurons to eliminate these regions entirely and assess if this replicates DPO's effect. As not all toxic neurons are exactly zero after DPO, we also apply activation patching to all toxic neurons with reduced positive activations in the pre-trained model, aligning their activations with post-DPO levels. Notably, many toxic neurons have small negative activations averaged across prompts due to the GELU activation function (see Appendix B). Therefore, we alternately exclude these neurons during ablation to avoid increasing toxicity (as seen in rows 2-4 in Table 1), focusing only on toxic neurons with positive activations.

Table 1 shows that, while ablating the most toxic neurons reduces toxicity to some degree, it falls short of the reduction achieved by DPO. Additionally, ablating over 1,000 toxic neurons significantly increases perplexity and degrades overall language quality. Similarly, activation patching does not achieve the same level of toxicity reduction as DPO.

Lee et al. [11] supported their claim by amplifying the top 7 toxic neurons' activations in the DPO-ed model, scaling their key vectors by a factor of 10, and reversed the toxicity (Table 1). However, we argue this intervention does not causally prove that dampening these neurons is DPO's primary mechanism. Amplifying these neurons by 10x drastically increases their impact beyond pre-DPO levels, likely to raise toxicity by boosting the norm of the toxic direction in the residual stream, similar to adding a steering vector [16]. This contrast echos the observation that inducing behaviour through unrealistic interventions on a model's component does not prove that this component alone is responsible for the behaviour [10], as seen in activation patching.

4.2 Computing neuron toxicity via projections

We follow [11] and assume that the toxic probe captures the aggregated toxicity feature direction in GPT2-medium. To track toxic reduction across neurons, we first compute the toxicity in MLP layers by projecting each layer's output onto the normalised probe direction. These projections are averaged across 1,199 prompts and 20 generated tokens. Figure 1a shows the projections before and after DPO, revealing a consistent drop in toxicity across layers.

We further decompose the reduction in MLP layer output projections (the gap between red and green lines in Figure 1a) into the sum of contributions from individual neurons in each layer. This decomposition is feasible because changes in layer activations equal the sum of the activation changes of individual neurons in that layer (see Equation 3 in Appendix A). Specifically, for neuron i, its contribution to toxicity reduction is computed as:

$$\text{toxic\_reduction}_i = (m_i^{pre} v_i^{pre} - m_i^{dpo} v_i^{dpo}) \cdot \frac{v_{toxic}}{|v_{toxic}|}, \tag{1}$$

where m_i^{pre} and m_i^{dpo} are the scalar activation coefficients of neuron i's value vector before and after DPO, and v_i^{pre} and v_i^{dpo} denote the corresponding value vectors. This equation captures the change in a neuron's toxicity projection following DPO.

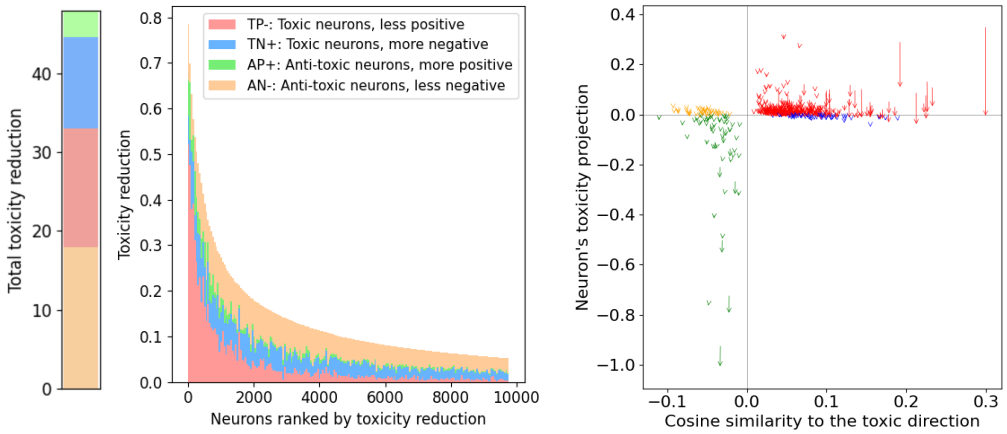
Our approach, which identifies the toxic feature component embedded in each neuron via projection, assumes that each neuron contributes proportionally along its activated direction. This approach is inspired by prior work showing that the toxic probe direction promotes toxic tokens when projected into the vocabulary space [11], with neurons acting as basis dimensions to increase the likelihood of these tokens [11, 7]. Additionally, this approach assumes that the toxicity probe direction remains unchanged after DPO following Lee et al. [11]. This assumption is supported by findings that sparse autoencoders (SAEs) trained on the base model can effectively reconstruct chat versions of models [12], suggesting that feature directions transfer well to safety fine-tuned models.

Figure 1b presents the cumulative sum of toxicity reduction contributed by all neurons in the model, ranked from most positive to negative projections. Interestingly, although over half of the neurons reduce the writing in the toxic direction after DPO, the remaining neurons actually increase it, forming an inverted U-shape curve. The peak total toxicity reduction is 48.0, which then declines to a net value 7.2 due to neurons adding toxicity. This highlights that DPO's minimal weight adjustments [11] accumulate to create noisy changes in neuron activations: some neurons reduce toxicity, while others increase it, reflecting a trade-off as DPO adjusts weights to generate non-toxic outputs [14]. Despite that reduced toxicity writing in some neurons may come at the expense of increased writing in others, the overall effect remains a net reduction in toxicity.

4.3 Identifying neuron groups for toxicity reduction

We identify four neuron groups that contributed the total toxicity reduction (48.0), and calculated each group's effect by summing their neuron contributions: toxic neurons activated less positively (TP_-), anti-toxic neurons activated less negatively (AN_-), toxic neurons activated more negatively (TN_+) and anti-toxic neurons activated more positively (AP_+). Here, a neuron is considered toxic or anti-toxic based on the cosine similarity between its value vector and the toxic probe direction. Note that group TP_- is the group Lee et al. [11] identified. Group TP_- and AN_- represent reduced writing in the toxic direction, while group TN_+ and AP_+ indicate proactive anti-toxic writing.

Figure 2a shows that TP_- and AN_- contribute the most to toxicity reduction, accounting for 31.8% and 37.3% of the total reduction, respectively. This means a sum of 69.1% of the reduction is due to erasing existing toxicity, while the remaining 30.9% comes from promoting anti-toxicity. Figure 2c shows the balanced contributions from four groups in the top 500 neuron contributors. Figure 2b shows that among the top neuron contributors, while TP_- initially dominates the distribution, the impact of AN_- grows with neuron ranks, with later neurons adding up more effects for toxicity reduction (see Appendix C for details). Figure 3 shows that per-layer toxicity reduction across neuron groups peaks in the later layers, mainly driven by TP_- and AN_-. This shows that DPO's most significant effects occur in later layers, consistent with [9]. These results show that toxicity reduction in DPO is a collective effort, with no single group driving the process alone. Instead, DPO's parameter changes accumulate to make small activation adjustments across neuron groups, both erasing toxicity and promoting anti-toxicity, resulting in a substantial overall effect.



(a) (b) Toxicity reduction by neuron groups. (c) Shifts in toxicity level by neuron groups.

Figure 2: **Contributions of four neuron groups to toxicity reduction.** (a) Proportions of toxicity reduction by each neuron group; (b) Stacked distribution of each group's contribution among the top 10000 neurons ranked by contribution. TP$_-$ initially dominates, with AN$_-$ gradually catching as neuron rank progresses; (c) Shifts in toxicity projection for the top 500 neurons ranked by contribution. Each arrow represents a neuron's projection change from pre-DPO to post-DPO levels, with all neurons shift with reduced toxicity.

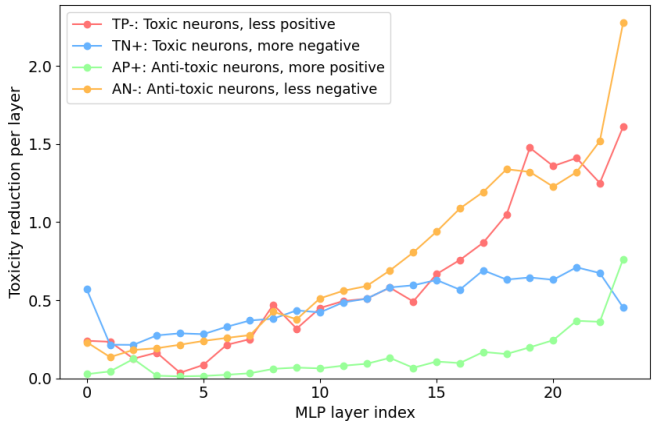


Figure 3: **Per-layer toxicity reduction by neuron groups.** DPO's parameter changes lead to the most significant toxicity reduction in the later layers, driven by TP$_-$ and AN$_-$.

5 Activation patching

To validate the effects of neuron groups on actual toxicity scores and link toxic feature reduction to changes in these scores, we apply activation patching to each group on the pre-trained model, adjusting their activations to post-DPO levels, and measure toxicity scores using the Perspective API.

Specifically, we apply averaged activation patching by assigning each neuron its mean post-DPO activation value (averaged across all prompts and 20 generated tokens) at the final token position for each prompt, guiding the generation of each next token. We acknowledge that using prompt-specific activation values for patching, rather than averaged values, can offer a better approximation of DPO's effects, but we do not pursue this due to computational constraints.

Table 2 shows that patching the three top-contributing neuron groups individually (TP$_-$, AN$_-$, TN$_+$) reduces toxicity. While no single group replicates DPO's full effect, patching the top two groups (TP$_-$, AN$_-$) achieves toxicity levels close to DPO, and patching the top three or all four groups yields reductions surpassing DPO, supporting the collaborative role of neuron groups in toxicity reduction. We also observe that general language capabilities are only minorly affected by

Table 2: **Toxicity, Perplexity (PPL) and F1 scores after patching on each neuron group**. Patching each top three contributing neuron groups individually reduces toxicity, while patching the top three or all four groups together achieves a toxicity reduction surpassing DPO.

Model	Intervention	Toxicity	PPL	F1
GPT2	None	0.453	21.70	0.193
GPT2	Patch TP_- neurons to post-DPO activations	0.335	21.69	0.190
GPT2	Patch TN_+ neurons to post-DPO activations	0.413	21.71	0.190
GPT2	Patch AN_- neurons to post-DPO activations	0.410	21.80	0.193
GPT2	Patch AP_+ neurons to post-DPO activations	0.455	21.72	0.193
GPT2	Patch TP_- and AN_- to post-DPO activations	0.239	21.78	0.189
GPT2	Patch TP_-, AN_-, TN_+ to post-DPO activations	0.193	21.76	0.174
GPT2	Patch all four groups to post-DPO activations	0.114	21.76	0.171
DPO	None	0.208	23.34	0.195

patching, as indicated by stable perplexity scores, suggesting that DPO subtly adjusts activations (unlike ablation or scaling in Table 1) to effectively preserve overall model performance.

6 Discussion

Our findings show that DPO's parameter changes do not just accumulate to dampen toxic neurons, but to reduce writing in the toxic feature direction by introducing subtle activation deviations from the feature across four neuron groups. In particular, changes in negative activations induced by GeLU are a significant source of these activation deviations (TN_+ and AN_-). These minor deviations in activation across neuron groups, especially in later layers, accumulate to reduce the overall writing of the toxic feature, resulting in decreased toxic generation. This understanding of DPO could motivate targeted interventions to replicate its effects. For example, could directly stripping out the toxic feature direction from the MLP weight matrix reduce toxic outputs? Future work could also explore the trade-offs in toxicity across neurons, disentangling how neurons counterbalance each other through weight adjustments and identifying the specific directions of these interactions.

For limitations, we recognise that focusing on a single linear probe direction to capture aggregated toxicity information following [11], may overlook nuanced aspects of toxicity. Different types of toxicity may manifest in various directions, representing distinct toxic behaviours (e.g., gender bias, curse words) or distributed as a toxic subspace spanning multiple neurons [17]. Future work could extend this study by exploring alternative methods to capture multiple toxic feature directions, such as using singular value decomposition (SVD) vectors derived from contrastive data pairs [17], and examining their patterns across neurons. Additionally, when computing neuron toxicity, we used projection to identify the portion of the toxic feature embedded in each neuron, assuming each neuron contributes proportionally to its activated direction. However, toxic features may actually be distributed across neurons in a more complex linear composition with varying weights. Alternative methods for decomposing features across neurons, such as sparse autoencoders (SAEs) [1], could be explored to track more fine-grained toxic feature changes across neurons.

7 Conclusion

This paper decodes DPO's mechanism by tracking how writing in a toxic feature direction, extracted by a linear probe, is reduced across MLP neurons. We challenge the prior explanation that DPO reduces toxicity primarily by dampening the most toxic activations [11]. By ablating or patching the most toxic neurons, we observe higher toxicity than with DPO, suggesting that this explanation is incomplete. Projecting neuron activations onto the toxic probe reveals that only 31.8% of the reduction comes from dampened toxic neurons. Instead, DPO achieves toxicity reduction through cumulative effects across four neuron groups with minor activation changes, both erasing toxicity and promoting anti-toxicity in the residual stream, resulting in reduced toxic feature writing. These group effects are validated through activation patching. Additionally, DPO introduces noisy adjustments to neuron activations, with some neurons increasing toxicity, suggesting that DPO functions as a balancing process across opposing neuron effects to achieve overall toxicity reduction.

References

[1] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.

[2] cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, nithum, and Will Cukierski. Toxic comment classification challenge, 2017. URL `https://kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge`. Accessed: 8-Nov-2024.

[3] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation, 2020. arXiv: 1912.02164.

[4] Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, Shrimai Prabhumoye, Alan W Black, Alexander Rudnicky, Jason Williams, Joelle Pineau, Mikhail Burtsev, and Jason Weston. The second conversational intelligence challenge (convai2), 2019. arXiv: 1902.00098.

[5] Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. Bias and fairness in large language models: A survey, 2024. arXiv: 2309.00770.

[6] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models, 2020. arXiv: 2009.11462.

[7] Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space, 2022. arXiv: 2203.14680.

[8] Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P. Dick, Hidenori Tanaka, Edward Grefenstette, Tim Rocktäschel, and David Scott Krueger. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks, 2023. arXiv: 2311.12786.

[9] Samyak Jain, Ekdeep Singh Lubana, Kemal Oksuz, Tom Joy, Philip H. S. Torr, Amartya Sanyal, and Puneet K. Dokania. What makes and breaks safety fine-tuning? a mechanistic study, 2024. arXiv: 2407.10264.

[10] Georg Lange, Alex Makelov, and Neel Nanda. An interpretability illusion for activation patching of feedforward mlp layers in language models, 2023. URL `https://www.alignmentforum.org/posts/RFtkRXHebkwxygDe2/an-interpretability-illusion-for-activation-patching-of`. Accessed: 8-Nov-2024.

[11] Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin Wattenberg, Jonathan K. Kummerfeld, and Rada Mihalcea. A mechanistic understanding of alignment algorithms: A case study on dpo and toxicity, 2024. arXiv: 2401.01967.

[12] Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2, 2024. arXiv: 2408.05147.

[13] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. arXiv: 1609.07843.

[14] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. arXiv: 2305.18290.

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. arXiv: 1707.06347.

[16] Daniel Tan, David Chanin, Aengus Lynch, Dimitrios Kanoulas, Brooks Paige, Adria Garriga-Alonso, and Robert Kirk. Analyzing the generalization and reliability of steering vectors, 2024. arXiv: 2407.12404.

[17] Rheeya Uppaal, Apratim Dey, Yiting He, Yiqiao Zhong, and Junjie Hu. Detox: Toxic subspace projection for model editing, 2024. arXiv: 2405.13967.

[18] Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. Assessing the brittleness of safety alignment via pruning and low-rank modifications, 2024. arXiv: 2402.05162.

[19] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's song in the ai ocean: A survey on hallucination in large language models, 2023. arXiv: 2309.01219.

A Mechanisms of MLP layers in Transformer

In this section, we provide details on MLP layers in transformer.

Each MLP layer l in a transformer processes the input $\mathbf{x}^l$ through two linear transformations with a point-wise activation function σ in between:

$$\text{MLP}^\ell(\mathbf{x}^\ell) = \sigma\left(W_K^\ell \mathbf{x}^\ell\right) W_V^\ell, \tag{2}$$

where $W_K^\ell, W_V^\ell \in \mathbb{R}^{d_{\text{mlp}} \times d}$, d_{mlp} and d are the dimensions of MLP layers and the residual stream, respectively. Expanding the equation gives:

$$\text{MLP}^\ell(\mathbf{x}^\ell) = \sum_{i=1}^{d_{\text{mlp}}} \sigma(\mathbf{x}^\ell \cdot \mathbf{k}_i^\ell)\mathbf{v}_i^\ell = \sum_{i=1}^{d_{\text{mlp}}} m_i^\ell \mathbf{v}_i^\ell. \tag{3}$$

Following Geva et al. [7] and Lee et al. [11], we refer to $\mathbf{k}_i^\ell$ (the i-th row of W_K^ℓ) as the *key vector*, and $\mathbf{v}_i^\ell$ (the i-th column of W_V^ℓ) as the *value vector* [7]. This equation shows that the MLP layer writes to the residual stream d_{mlp} times, once for each value vector $\mathbf{v}_i^\ell$ scaled by an activation coefficient m_i^ℓ. Geva et al. [7] showed that each sub-update $m_i^\ell \mathbf{v}_i^\ell$ promotes the likelihood of certain tokens to be generated. Our experiments used GPT2-medium, which consists of 24 layers, with $d = 1024$ and $d_{\text{mlp}} = 4096$.

B Most toxic neurons have negative activations

In this section, we explain why directly ablating the most toxic neurons leads to diminished toxicity reduction as more neurons are ablated, as seen in rows 2-4 in Table 1.

Figure 4 shows the average activations of the top 100 toxic neurons across all prompts and 20 generated tokens, both before and after DPO. Aside from the first few, most neurons are inactive and display small negative activations due to the GELU function. This suggests that simply zeroing their activations may inadvertently increase toxicity.

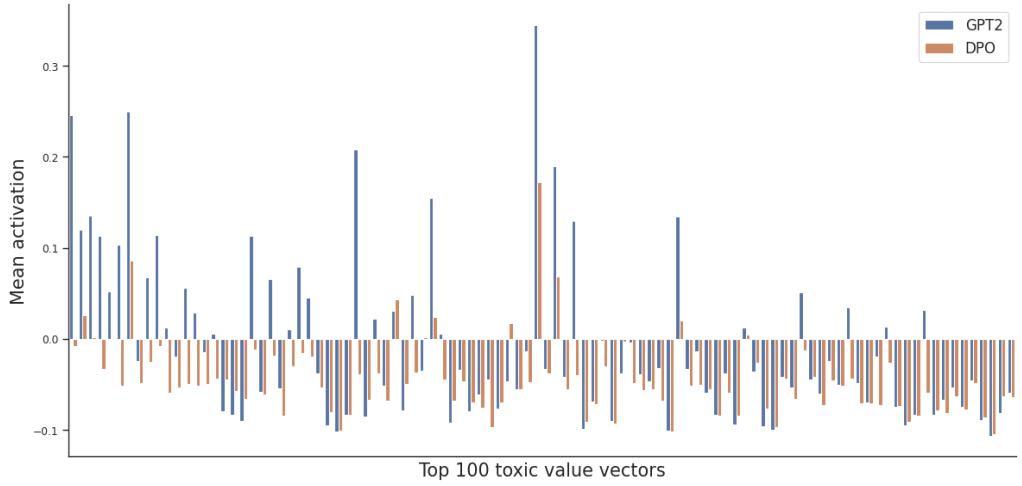
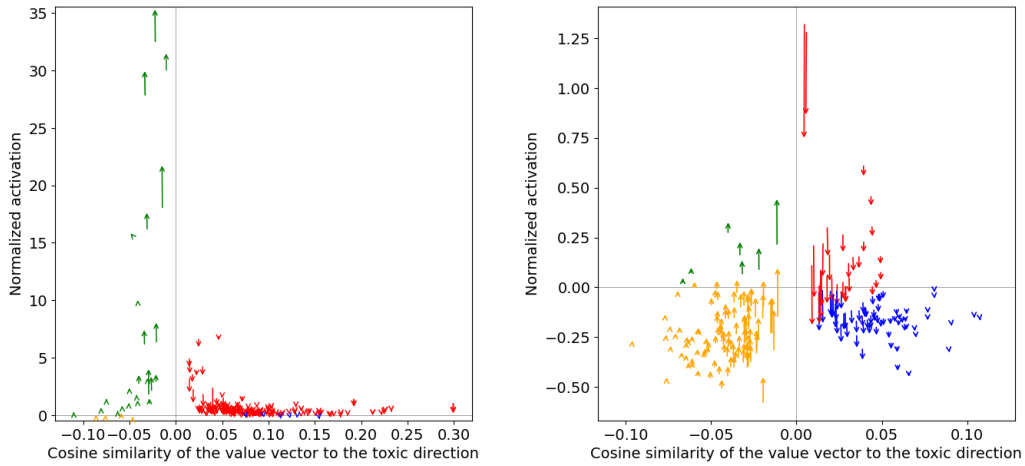


Figure 4: **Activations of the top 100 toxic neurons before and after DPO.** Most neurons have negative activations averaged across prompts, both before and after DPO.

C Toxicity reduction by neuron groups

In this section, we demonstrate how neuron groups contribute to toxicity reduction as neuron rank progresses.

Figure 5 compares the contribution of the most toxic neurons, focusing on the top 200 versus those ranked between 3000 and 3200. Initially, as shown in Figure 5a shows that initially TP_ constitutes the majority of the top 200 toxic neurons and dominates their contribution. However, further down the neuron ranks, as seen in Figure 5b, contributions from the other three neuron groups, particularly AN_, accumulate more effects and become more significant.



(a) Shifts in activations on top 200 toxic neurons. (b) Shifts in activations on top 3000-3200 toxic neurons.

Figure 5: **Shifts in activations of top toxic neurons by neuron groups.** (a) In the top 200 toxic neurons, the primary contributing group is TP_; (b) For toxic neurons ranked 3000-3200, contributions are more evenly distributed across all four groups.