# Toxic Neurons Aren't Enough to Explain DPO: A Mechanistic Analysis for Toxicity Reduction

**Yushi Yang**[*]
University of Oxford

**Filip Sondej**
Jagiellonian University

**Harry Mayne**
University of Oxford

**Adam Mahdi**
University of Oxford

## Abstract

Safety fine-tuning algorithms are widely used to reduce harmful outputs in language models. While studies show that these algorithms induce minimal changes to pre-trained model parameters, the mechanisms of how such small parameter changes lead to harm reduction remain unclear. When studying the direct preference optimization (DPO) algorithm for toxicity reduction, current explanation claims that DPO reduces toxicity by dampening activations of the most toxic MLP neurons. However, our activation patching experiments show that this explanation is incomplete. Projections onto a toxicity probe show that only 4.9% of toxicity reduction comes from dampened toxic neurons. Instead, DPO reduces toxicity through distributed activation shifts across four neuron groups: two removing toxicity and two promoting anti-toxicity, cumulatively shifting MLP outputs away from toxicity. Neurons that do not promote toxic tokens still contribute to this reduction through their weakly aligned components. These distributed activation shifts, induced from DPO's minimal parameter changes, form a mask over the pre-trained toxic capabilities, while being small enough to preserve model's general language capabilities. Building on these insights, we propose an activation patching technique on the identified neuron groups, outperforming DPO in reducing toxicity while maintaining general language capabilities. [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, 17]. To address these issues, researchers have developed safety fine-tuning algorithms, such as proximal policy optimization (PPO) [14] and direct preference optimization (DPO) [13], 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 model capabilities to generate undesirable outputs are not eliminated but hidden [10, 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, suggested that DPO reduces toxicity by dampening the activations of the most toxic MLP neurons to bypass the toxic regions they create in the residual stream [10]. Our study verifies this claim by analysing toxic feature writing along a probe direction across MLP neurons, revealing the explanation to be incomplete. Our findings are:

[*] 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 Socially Responsible Language Modelling Research (SoLaR).

- *Dampened toxic neurons account for a very limited effect of DPO.* We patch the most toxic neurons to post-DPO activations and find that this gives significantly higher toxicity levels than DPO, showing that dampened toxic neurons alone [10] account for a very limited amount of DPO's effect. By projecting onto the toxicity probe, we show that dampened toxic neurons account for just 4.9% of the total toxicity reduction.
- *DPO reduces toxicity through distributed activation shifts across four neuron groups.* DPO achieves toxicity reduction by engaging over half of the neurons, which we categorise into four groups, two reducing toxicity and two promoting anti-toxicity. These groups collectively shift MLP layer outputs away toxicity. Interestingly, the rest of neurons modified by DPO actually increase toxicity, indicating that DPO creates a trade-off in toxicity.
- *Patching the identified neuron groups outperforms DPO.* Leveraging these findings, we apply activation patching to the identified neuron groups responsible for toxicity reduction, which surpasses DPO in reducing toxicity while preserving general language capabilities.

## 2 Related work

**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. [16] demonstrated the brittleness of safety fine-tuning methods, showing that pruning just 3% of targeted model parameters can unlock the model from aligned behaviours. These studies suggest that fine-tuning algorithms minimally alter model parameters to achieve the desired outputs.

In our reference study, Lee et al. [10] studied how the DPO algorithm works internally to reduce toxicity. Referring to the second weight vector for an MLP neuron as the *value vector* [7] (Appendix A), Lee et al. [10] proposed that DPO reduces toxicity by dampening the activations on most toxic MLP neurons, whose value vectors project onto toxic tokens and align the most with a toxicity linear probe, thereby shifting model activations out of toxic regions associated with them. Our study tests this claim and finds it incomplete, as discussed below.

## 3 Experimental setup

To test Lee et al. [10]'s claims and ensure fair comparisons, we replicate their experimental setup, including the same language model, dataset of toxicity-eliciting prompts, probe training method, and evaluation metrics.

**Model and dataset** We focus on GPT-2 medium with 355M parameters and 24 layers, a residual stream dimension of 1024, and an MLP hidden layer dimension of 4096 [10]. GPT-2 Medium is well-suited for studying toxicity reduction, as it is easier to elicit toxic responses with designed prompts compared to larger base models, making it ideal for capturing toxicity changes. We use the DPO-ed version of GPT2-medium [10] fine-tuned on 24,576 pairs of toxicity data generated by PPLM pipeline [3] on Wikitext-2 prompts. To elicit toxic outputs, we use the "challenge" subset of REALTOXICITYPROMPTS [6], containing 1,199 highly toxic prompts.

**Toxicity probe** We train a linear probe at the last layer of GPT2-medium to classifiy toxic text from non-toxic text [10]. The toxicity linear probe $W_{toxic}$ was trained on a binary classification task using the Jigsaw toxic comment classification dataset (561,808 comments) [2]. The probe direction projects to toxic tokens and serves as an effective steering vector to reduce toxicity when applied to the final layer of GPT-2 Medium [10]. We follow Lee et al. [10] and view this probe as capturing the aggregated toxicity feature direction in GPT2-medium.

**Evaluation metrics** To evaluate toxicity and language quality of the generation, we measure *toxicity scores* by passing the generated responses through the Perspective API [7], *perplexity* by matching generated tokens on the Wikitext-2 dataset [12], and *F1 scores* on 2,000 Wikipedia sentences [4].

Table 1: **Toxicity, Perplexity (PPL) and F1 scores after activation patching on neurons**. Patching toxic neurons to post-DPO levels achieves limited toxicity reduction compared to DPO. In contrast, patching all four of our identified neuron groups surpasses DPO in reducing toxicity while preserving perplexity, validating their effects.

| **Model** | **Intervention** | **Toxicity** | **PPL** | **F1** |
|---|---|---|---|---|
| GPT2 | None | 0.453 | 21.70 | 0.193 |
| DPO | None | 0.208 | 23.34 | 0.195 |
| GPT2 | Patch the toxic neurons to post-DPO activations | 0.404 | 21.70 | 0.193 |
| GPT2 | Patch the positively activated toxic neurons | 0.409 | 21.70 | 0.193 |
| GPT2 | Ablate the toxic neurons | 0.409 | 22.13 | 0.192 |
| GPT2 | Ablate the positively activated toxic neurons | 0.398 | 21.74 | 0.193 |
| GPT2 | Patch the $\text{TP}_-$ group to post-DPO activations | 0.335 | 21.69 | 0.190 |
| GPT2 | Patch the $\text{TN}_+$ group to post-DPO activations | 0.413 | 21.71 | 0.190 |
| GPT2 | Patch the $\text{AN}_-$ group to post-DPO activations | 0.410 | 21.80 | 0.193 |
| GPT2 | Patch $\text{TP}_-$ and $\text{AN}_-$ to post-DPO activations | 0.239 | 21.78 | 0.189 |
| GPT2 | Patch $\text{TP}_-$, $\text{AN}_-$, $\text{TN}_+$ 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 |

## 4 Track toxicity reduction across neurons

### 4.1 Activation patching the toxic neurons

**Identify toxic neurons** Follow Lee et al. [10], we identify toxic neurons as those whose value vectors (the second weight vector for an MLP neuron) have the highest cosine similarity to $W_{\text{toxic}}$ and project onto toxic tokens in the vocabulary space. We narrow this down to $N = 128$ toxic value vectors, as Lee et al. [10] found this to be the number required to form a stable toxic space with singular value decomposition, beyond which adding more value vectors does not expand the toxic basis further. Also beyond $N = 128$, the value vectors no longer project onto toxic tokens (Table 2 in Appendix B).

**Patch toxic neurons** We apply activation patching to the 128 toxic neurons after DPO, aligning their activations to their post-DPO levels. Additionally, we patch the 36 out of 128 toxic neurons with positive activations before DPO to test if their dampened activations reduce toxicity. For a stronger intervention, we ablate these neurons (i.e., zero out their activations) to eliminate their associated toxic regions in the residual stream entirely.

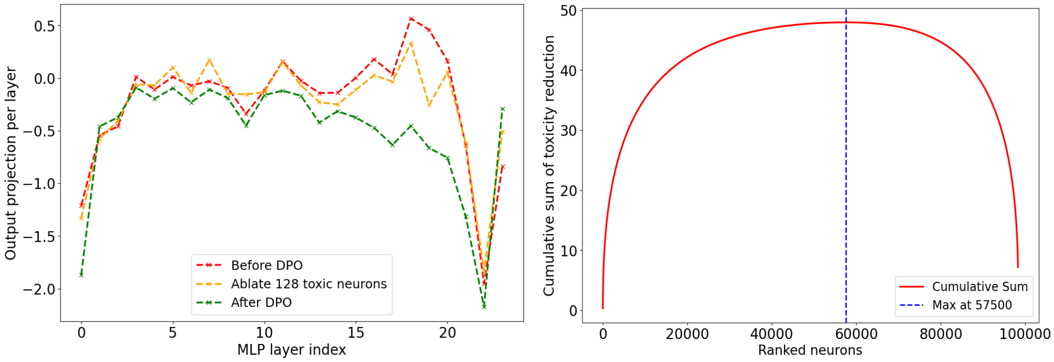
Table 1 shows that, while patching or ablating the toxic neurons reduces toxicity to some extent, it falls short to the reduction achieved by DPO. This suggests that dampened toxic neurons alone [10] account for a very limited amount of DPO's effect.

### 4.2 Compute neuron toxicity via projections

**MLP layer projection** Instead, we track toxicity reduction across MLP layers by projecting each MLP layer's output onto the normalised $W_{\text{toxic}}$ direction. These projections are averaged over layer activations for 1,199 prompts and 20 generated tokens. Figure 1a shows a consistent drop in toxicity projections across layers after DPO, which consistently outperforms ablating the toxic neurons.

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

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



(a) Projection of MLP layer outputs to toxicity probe. (b) Cumulative toxicity reduction by neurons.

Figure 1: **Toxicity projection to the probe across MLP layers and neurons.** (a) Output projections of MLP layers before DPO (red), after ablating the toxic neurons (yellow), and after DPO (green). (b) Cumulative sum of toxicity reduction contributed by neurons, with neurons ranked from highest to lowest toxicity reduction. The U-shaped curve shows that many neurons increase toxicity after DPO.

Here, $m_i^{pre}$ and $m_i^{dpo}$ are the activation coefficients (scalars) of neuron $i$'s value vector before and after DPO, while $v_i^{pre}$ and $v_i^{dpo}$ represent the value vectors. This equation quantifies the change in each neuron's toxicity projection after DPO. This equation assumes the toxicity probe direction remains unchanged after DPO following [10]. This assumption is supported by findings that sparse autoencoders (SAEs) trained on the base model can effectively reconstruct chat versions of models [11], suggesting that feature directions transfer well between pre-trained and safety fine-tuned models.

**Results** Following this equation, the summed toxicity reduction of the 128 toxic neurons with dampened positive activations accounts for only **4.9%** of the total toxicity reduction. Moreover, the U-shaped cumulative reduction in Figure 1b shows that, while DPO engages over half of the neurons to reduce toxicity, the remaining neuron activations are adjusted to **increase** toxicity. This suggests that DPO creates opposing neuron effects, where reduced toxicity in some neurons comes at the expense of increased writing in others, potentially an inevitable trade-off driven by noisy parameter updates.

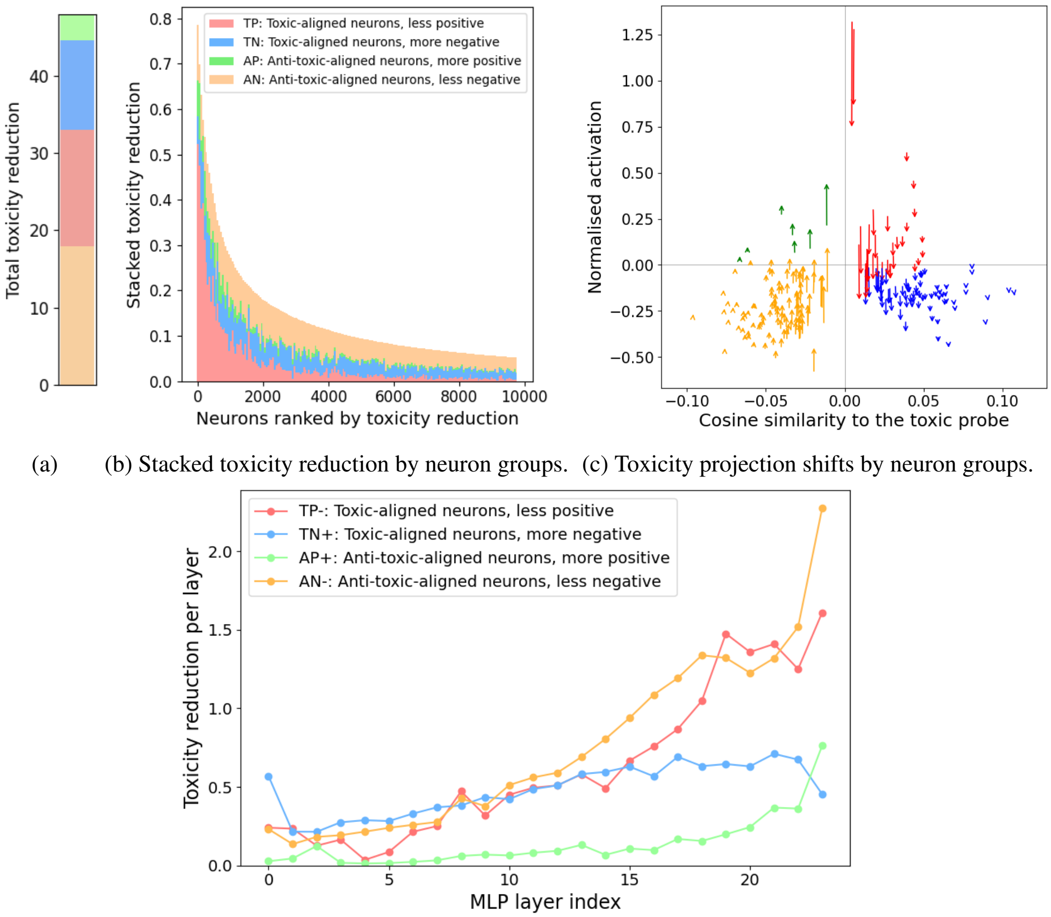
### 4.3 Categorise neuron groups for toxicity reduction

**Define neuron groups** Following neuron toxicity projection, we categorise neurons contributing to toxicity reduction (the rising part of curve in Figure 1b) into four groups based on two criteria: (1) whether they align positively or negatively with the toxicity probe, and (2) whether their activations are increased or decreased. The groups are:

- Toxic-aligned neurons activated less positively ($\text{TP}_-$);
- Toxic-aligned neurons activated more negatively ($\text{TN}_+$);
- Anti-toxic-aligned neurons activated less negatively ($\text{AN}_-$);
- Anti-toxic-aligned neurons activated more positively ($\text{AP}_+$).

Here, a neuron is considered toxic-aligned or anti-toxic-aligned based on whether the cosine similarity between its value vector and the toxic probe direction is positive or negative. Many neurons that do not project directly to toxic tokens (Appendix B) are still weakly aligned with the probe. In particular, groups $\text{TN}_+$ and $\text{AN}_-$ arise because the GELU activation function allows small negative activations on neurons (Appendix C). Overall, $\text{TP}_-$ and $\text{AN}_-$ represent reduced writing in the toxic direction, while $\text{TN}_+$ and $\text{AP}_+$ reflect proactive anti-toxic writing.

**Analyse neuron groups** Figure 2 shows that the contributions of different neuron groups to toxicity reduction. Figure 2a shows that $\text{TP}_-$ and $\text{AN}_-$ are the primary contributors, accounting for 31.8% and 37.3% of the total reduction, respectively. Together, these groups contribute 69.1% of the



(a) (b) Stacked toxicity reduction by neuron groups. (c) Toxicity projection shifts by neuron groups.

(d) Toxicity reduction per layer by neuron groups.

Figure 2: **Toxicity reduction contributed by four neuron groups.** (a) Proportions of toxicity reduction by neuron groups; (b) Stacked contribution by neuron groups among the top 10,000 contributing neurons. TP$_{-}$ initially dominates, with AN$_{-}$ gradually catching as neuron rank progresses; (c) Shifts in toxicity projections for the top 3,000–3,200 contributing neurons, with the four-quadrant separation defining the groups. Each arrow indicates a neuron's projection change from pre-DPO to post-DPO levels, all showing reduced toxicity; (d) Per-layer toxicity reduction by neuron groups. Four neuron groups cumulatively reduce toxicity across layers, with the most significant reductions occurring in the later layers, primarily driven by TP$_{-}$ and AN$_{-}$.

reduction by removing existing toxicity, while the remaining 30.9% results from promoting anti-toxicity. Figure 2b shows that among the top neuron contributors, TP$_{-}$ initially dominates, but the influence of AN$_{-}$ grows as neuron rank increases, with later neurons adding more effects to toxicity reduction (Appendix D). Figure 2c illustrates the balanced contributions from the four groups among the top 3,000–3,200 neuron contributors, with the four-quadrant separation visually representing the definitions of the groups. Figure 2d shows that the four neuron groups cumulatively reduce toxicity layer by layer, with peak reductions occurring in the later layers, primarily driven by TP$_{-}$ and AN$_{-}$. In summary, these results show that toxicity reduction in DPO is a collective effort across all four groups, with their activation shifts collectively steering MLP layer outputs away from toxicity.

**Validate neuron groups** To validate the effects of neuron groups on actual toxicity scores, we perform activation patching for each group, adjusting their activations to post-DPO levels, and measure toxicity scores using the Perspective API. Specifically, we apply averaged patching, 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.

Table 1 shows that patching each of the top three contributing neuron groups individually ($\mathrm{TP}_-$, $\mathrm{AN}_-$, $\mathrm{TN}_+$) reduces toxicity. Patching the top two groups ($\mathrm{TP}_-$ and $\mathrm{AN}_-$) achieves toxicity levels close to those of DPO, while patching the top three or all four groups results in reductions surpassing DPO. This validates the contribution of each neuron group and supports our claim regarding their collaborative role in toxicity reduction. The superior performance of patching all four groups may result from excluding neurons that increase toxicity (the falling part of curve in Figure 1b). Additionally, perplexity scores remain minimally affected by patching, indicating that the distributed small activation shifts induced by DPO across the four neuron groups effectively preserve general language capabilities.

## 5 Discussion

**Mechanisms of DPO** Our study challenges the monosemantic view of neuron roles proposed by Lee et al. [10], which attributes toxicity to a few specific toxic neurons. Our findings reveal that DPO's parameter changes do not simply dampen toxic neurons but instead reduce writing in the toxic direction through subtle activation shifts across four neuron groups. These shifts accumulate within each layer, with per-layer changes peaking in the later layers, collectively decreasing the overall writing of toxicity. Since DPO minimally alters model parameters [10], the pre-trained toxic capabilities remain largely intact. Instead, the distributed activation shifts induced by DPO's minimal parameter changes form a mask over the toxic capabilities, effectively hiding them while reducing toxicity and remaining small enough to preserve the model's general language capabilities.

**Limitations** Our work relies on a single linear probe direction to capture aggregated toxicity information in the model, which may overlook more nuanced aspects of toxicity. For instance, different toxic behaviours, such as gender bias or the use of curse words, may manifest in different directions. Future work could extend this approach by computing projections onto a toxic subspace spanned by multiple toxic feature directions [15], to identify more accurate neuron groups. Additionally, for computing neuron toxicity, we used projection to estimate the portion of the toxic feature embedded in each neuron, assuming proportional contribution based on each neuron's activated direction. However, toxic features may actually be distributed across neurons in a more complex linear composition with varying weights. Alternative methods, such as sparse autoencoders (SAEs) [1], could be explored to track fine-grained changes in toxic features across neurons. Future work could also investigate the toxicity trade-offs observed in neurons, disentangling how adjustments in neuron weights counterbalance one another to create these trade-offs.

## 6 Conclusion

This study decodes DPO's mechanism by analysing how writing in a toxic feature direction, identified via a linear probe, is reduced across MLP neurons. We challenge the prior explanation that DPO reduces toxicity by dampening the most toxic neurons [10], demonstrating through activation patching that this explanation is incomplete. Projecting neuron activations onto the toxic probe reveals that only 4.9% of the total reduction is due to dampened toxic neurons. Instead, DPO reduces toxicity through cumulative effects across four neuron groups, progressively reducing toxic feature writing across layers. These distributed activation shifts form a mask over the pre-trained toxic capabilities while remaining small enough to preserve the model's general language capabilities. Based on these findings, we propose an activation patching intervention targeting the identified neuron groups, which outperforms DPO in reducing toxicity while maintaining general language capabilities.

# 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] 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.

[11] 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.

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

[13] 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.

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

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

[16] 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.

[17] 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 Transformers

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 $\sigma$ 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. [10], we refer to $\mathbf{k}_i^{\ell}$ (the $i$-th row of $W_K^{\ell}$) as the *key vector*, and $\mathbf{v}_i^{\ell}$ (the $i$-th column of $W_V^{\ell}$) as the *value vector* [7]. This equation shows that the MLP layer writes to the residual stream $d_{\text{mlp}}$ times, once for each value vector $\mathbf{v}_i^{\ell}$ scaled by an activation coefficient $m_i^{\ell}$. 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.

# B Projections of value vectors to the vocabulary space

In this section, we present how value vectors project onto tokens as their cosine similarity with the toxicity probe decreases.

Table 2 shows that value vectors cease projecting onto toxic tokens around $N = 60$, and by $N = 128$, they no longer project onto toxic tokens at all.

Table 2: Top tokens that value vectors projected onto in the vocabulary space.
WARNING: THESE EXAMPLES ARE HIGHLY OFFENSIVE.

| VECTOR | RANK | TOP TOKENS |
|---|---|---|
| $\mathbf{W}_{\text{Toxic}}$ | - | c*nt, f*ck, a**hole, d*ck, wh*re, holes |
| MLP.$\mathbf{v}_{19}^{770}$ | 1 | sh*t, a**, cr*p, f*ck, c*nt, garbage, trash |
| MLP.$\mathbf{v}_{18}^{2669}$ | 3 | degenerate, whining, idiots, stupid, smug |
| MLP.$\mathbf{v}_{3}^{3680}$ | 10 | se*ist, feminist, Femin, femin, misogyn |
| MLP.$\mathbf{v}_{13}^{253}$ | 18 | c*m, d*ck, icles, icle, bo*bs, naughty, |
| MLP.$\mathbf{v}_{7}^{3358}$ | 29 | cr*p, whine, sh*t, uphem, shri, bullsh*t |
| MLP.$\mathbf{v}_{6}^{3972}$ | 44 | death, extermination, Corpse, decap, torture |
| MLP.$\mathbf{v}_{6}^{3972}$ | 50 | f*cking, d*mn, sinful, hell, immoral |
| MLP.$\mathbf{v}_{19}^{1438}$ | 56 | c*m, c*ck, orgasm, missionary, anal |
| MLP.$\mathbf{v}_{19}^{1438}$ | 59 | burdens, bad, offending, imped, horrible |
| MLP.$\mathbf{v}_{19}^{1438}$ | 60 | Bench, rodu, Sequ, RIP, Brist, Vers |
| MLP.$\mathbf{v}_{19}^{1438}$ | 61 | uple, buff, virgin, intangible, nw, illiter |
| MLP.$\mathbf{v}_{19}^{1438}$ | 128 | vous, ccoli, weed, ername, Timber, alyses |
| MLP.$\mathbf{v}_{19}^{1438}$ | 129 | roma, ocker, oley, hiba, osure, urden |
| MLP.$\mathbf{v}_{19}^{1438}$ | 130 | revolving, Bree, Hoo, dise, Cheong, uay |

# C Most toxic neurons have negative activations

In this section, we explain why we focus on patching or ablating only toxic neurons with positive activations in Section 4.1, as many toxic value vectors have small negative activations, and ablating them directly may increase toxicity.

Figure 3 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.

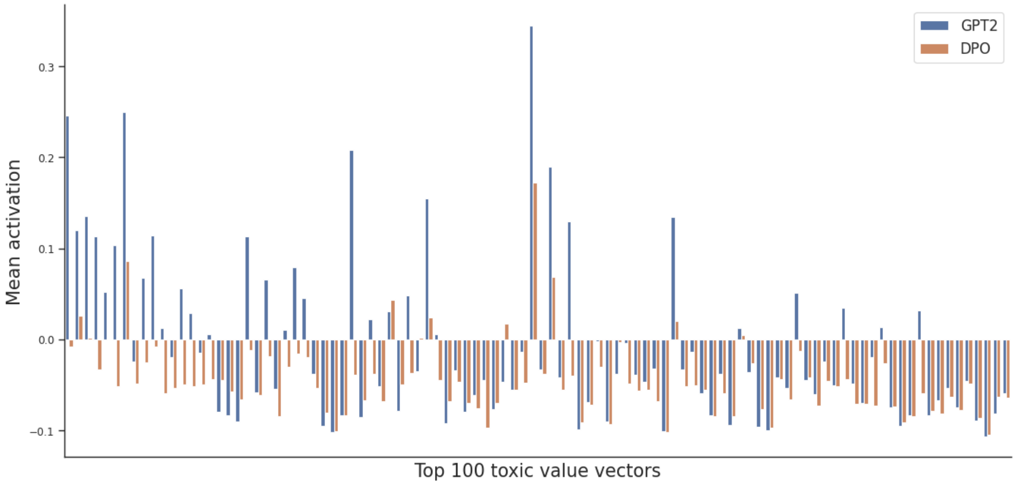
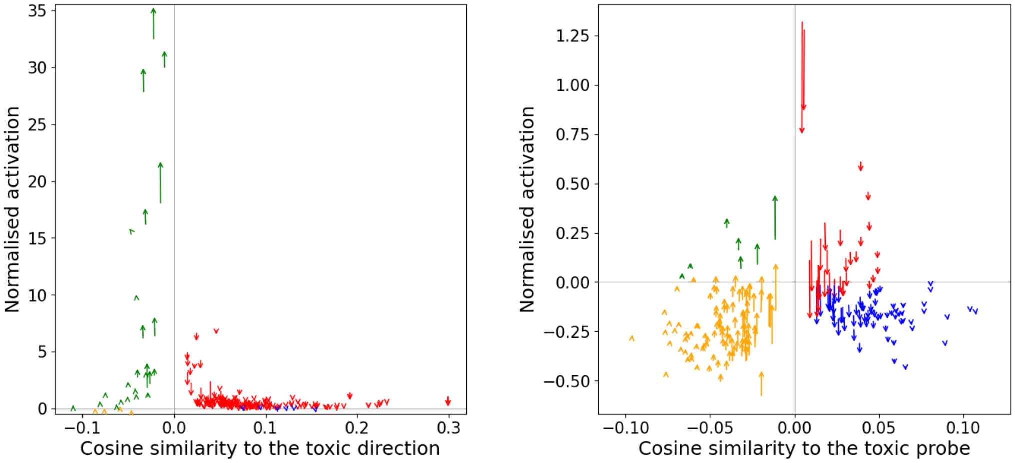


Figure 3: **Activations of the top 100 toxic neurons before and after DPO.** Most neurons have negative activations averaged across prompts, both before and after DPO.

# D Toxicity reduction by neuron groups

In this section, we demonstrate how neuron groups contribute to toxicity reduction as neuron rank progresses, further illustrating Figure 2b.

Figure 4 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 4a 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 4b, 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 4: **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.