How Does DPO Reduce Toxicity? A Mechanistic Neuron-Level Analysis

Anonymous ACL submission

Abstract

Safety fine-tuning algorithms reduce harmful outputs in language models, yet their mechanisms remain under-explored. Direct Preference Optimization (DPO) is a popular choice of algorithm, but prior explanations—attributing its effects solely to dampened *toxic neurons* in the MLP layers—are incomplete. In this study, we analyse four language models (Llama-3.1-8B, Gemma-2-2B, Mistral-7B, GPT-2-Medium) and show that toxic neurons only account for 2.5% to 24% of DPO's effects across models. Instead, DPO induces distributed activation shifts across all MLP neurons to create a net toxicity reduction. We attribute this reduction to four neuron groups—two aligned with reducing toxicity and two promoting anti-toxicity—whose combined effects replicate DPO across models. To further validate this understanding, we develop an activation editing method that mimics DPO through distributed shifts along a toxicity representation. This method outperforms DPO in reducing toxicity while preserving perplexity, without requiring any weight updates. Our work provides a mechanistic understanding of DPO and introduces an efficient, tuning-free alternative for safety fine-tuning. Our code is available in the anonymous repository: anonymous.4open.science/r/dpo-mlp-toxic.

1 Introduction

The growing capabilities of large language models (LLMs) also lead to the encoding of undesirable behaviours (Gehman et al., 2020; Gallegos et al., 2024). To mitigate harmful outputs, researchers have developed fine-tuning algorithms to prioritise human-preferred responses through reward modelling (Schulman et al., 2017; Shao et al., 2024). Among these, Direct Preference Optimization (DPO) has been a popular algorithm given its simplicity that directly optimises the policy model (Rafailov et al., 2024). While such methods effectively reduce harmful behaviours at the output level, there is limited mechanistic understanding of how they achieve this internally. This gap limits our ability to explain their vulnerability to jailbreaks and adversarial fine-tuning (Wei et al., 2023; Yang et al., 2023; Qi et al., 2023).

Recent studies found that fine-tuning algorithms lead to superficial changes, allowing models to retain the undesirable capabilities (Jain et al., 2024; Yang et al., 2023). In particular, Lee et al. (2024) suggested that DPO reduces toxicity by dampening the activations of a few *toxic neurons* in the MLP layers. While this offers an intuitive explanation, it assumes that toxicity is localised to a small subset of neurons—a strong claim that may oversimplify how safety fine-tuning works. In this paper, we show that this explanation is incomplete, and offer a more comprehensive analysis of DPO's mechanism across four LLMs: Llama-3.1-8B, Gemma-2-2B, Mistral-7B and GPT-2-Medium.

Toxic neurons are not enough to explain DPO. Namely, we use activation patching to isolate the role of toxic neurons, and observe only a partial drop in toxicity across models (2.5% to 24%) compared to DPO. Where, then, does the rest of DPO's toxicity reduction come from?

Four neuron groups reduce toxicity. We show that DPO induces more nuanced, distributed activation shifts across all MLP neurons than previously suggested. We identify four mutually exclusive neuron groups that consistently contribute to toxicity reduction across models. Their post-DPO activation changes depend on their orientation relative to the toxicity representation. Using activation patching, we show that their combined influence can match or even exceed the toxicity reduction achieved by DPO.

Activation editing to replicate DPO. To validate our understanding, we develop a simple activation editing method to replicate DPO. Unlike the previous post-hoc patching analyses, our method does not rely on access to post-DPO activations, nor does it require weight updates or pairwise preference data. Instead, we leverage our observations to edit activations based on the orientation of MLP weights relative to a toxicity representation. This method consistently outperforms DPO across models, showing that DPO-like effects can be achieved with minimal intervention and without fine-tuning.

2 Related Work

Here we review the DPO algorithm, the Transformer MLP layers and related work on mechanisms of safety fine-tuning algorithms.

DPO algorithm. DPO is a fine-tuning algorithm designed to align LLMs with pairwise human preference data (Rafailov et al., 2024). Given pairwise preference data

$$\left\{\left(x^{(i)}, y_+^{(i)}, y_-^{(i)}\right)\right\}_{i=1}^N,$$

where x is the input prompt, y_+, y_- are pairwise preferred and non-preferred continuations, DPO fine-tunes a policy model $\pi_\theta(y_+ \mid x)$ that assigns a higher likelihood to $y_+^{(i)}$ compared to $y_-^{(i)}$.

The DPO loss is defined as:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\log\sigma\left(\beta\left(r_\theta(x, y^+) - r_\theta(x, y^-)\right)\right),$$

where σ is the sigmoid function, β is a temperature hyperparameter and r_θ is the derived reward regularised using the reference model π_{ref}, that is

$$r_\theta(x, y) = \log\frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)}.$$

MLP layers. MLPs apply two linear transformations with a non-linearity σ in between:

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

where $W_K^\ell, W_V^\ell \in \mathbb{R}^{d_{\text{mlp}} \times d}$, d_{mlp} and d are the dimensions of MLP hidden layers and the residual stream. MLPs can be re-expressed as:

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

where $\mathbf{k}_i^\ell, \mathbf{v}_i^\ell \in \mathbb{R}^d$ are the i-th row of W_K^ℓ and W_V^ℓ, respectively. For each MLP neuron i, we refer to $\mathbf{v}_i^\ell$ as its *value vector*, following Geva et al. (2022) and Lee et al. (2024). The scalar $m_i^\ell \in \mathbb{R}$ is an *activation score* that controls the scaling of the value vector $\mathbf{v}_i^\ell$. This means an MLP layer writes to the residual stream d_{mlp} times, once per neuron, via the activation-weighted value vector $m_i^\ell \mathbf{v}_i^\ell$.

Recent models (Llama, Gemma, Mistral) replace MLPs with Gated Linear Units (GLUs) (Shazeer, 2020). GLUs can similarly be expressed as a weighted sum of its value vectors as in (1), where each weight is determined by some non-linear activation. See Appendix A for details.

Mechanisms of safety fine-tuning algorithms. Recent studies have shown that fine-tuning induces superficial weight changes, leaving most pretrained capabilities intact. Jain et al. (2023) found that fine-tuning on synthetic tasks produces 'wrappers', i.e. localised weight changes in later layers optimised for each task. Qi et al. (2024) found that aligned models primarily adapt their generative distribution in the first few output tokens. Wei et al. (2024) showed that pruning just 3% of targeted parameters can undo safety alignment, highlighting the brittleness of safety mechanisms. These findings suggest that safety fine-tuning reduces harmful outputs through subtle, targeted weight changes rather than large-scale rewiring.

Lee et al. (2024) studied the mechanisms of how DPO reduces toxic outputs, attributing its effects to dampened activations of a few toxic MLP value vectors. We revisit this claim and find it to be incomplete, as shown in Section 4.

3 Experiment Setup

Here we describe the tools used in this study, including the data and models, linear probes, projections and activation patching.

3.1 Data and Models

Toxicity-eliciting prompts. We use the 'challenge' subset (N=1,199) of *RealToxicityPrompts* (Gehman et al., 2020) to elicit toxic outputs from each model. This subset is designed to trigger extremely toxic completions, making it a strong testbed for evaluating safety fine-tuning.

Models. We study four pre-trained LLMs: Llama-3.1-8B (Grattafiori et al., 2024), Gemma-2-2B (Riviere et al., 2024), Mistral-7B (Jiang et al., 2023), and GPT-2 Medium (Radford et al., 2019). GPT-2 Medium is included to compare with claims made in Lee et al. (2024). We generate toxic outputs from each LLM using greedy decoding. Appendix B provides the MLP specification for each model.

Evaluation metrics. We report three metrics: *toxicity scores* using Detoxify (Hanu, 2020), a BERT model fine-tuned for toxicity classification that assigns a likelihood score of a text being toxic; *log perplexity*, the average negative log-likelihood of generated tokens on the Wikitext-2 dataset (Merity et al., 2016); *F1 scores*, the harmonic mean of precision and recall based on token overlap across 2,000 Wikipedia sentences (Lee et al., 2024). The latter two metrics measure general language quality, where F1 complements perplexity by capturing exact token matches.

DPO training. We implement DPO using 24,576 toxicity contrastive pairs generated from Wikitext-2 prompts (Lee et al., 2024). See Appendix C for training hyperparameters.

3.2 Per-Neuron Toxicity Contributions

We measure per-neuron contributions to toxicity by projecting activations onto linear toxicity probes. We describe how we extract these probes, validate their effects and compute per-neuron contributions.

Linear probes. To extract toxicity representations, we train linear probes W_{Toxic} to classify toxic versus non-toxic inputs for each model. The probe is trained on the final-layer residual stream $\bar{\mathbf{x}}^{L-1}$, averaged across all token positions:

$$P(\text{toxic} \mid \bar{\mathbf{x}}^{L-1}) = \sigma(W_{\text{Toxic}}\bar{\mathbf{x}}^{L-1} + b),$$

where σ is the sigmoid function, $W_{\text{Toxic}} \in \mathbb{R}^d$ is the learned probe vector. We use the *Jigsaw Toxic Comment Classification* dataset (cjadams et al., 2017), which contains 561,808 comments labelled as toxic or non-toxic.

Across all four models, the linear probes achieve over 91% test accuracy using a 90:10 train/test split (Appendix Table 11). When projected onto each model's vocabulary space via the unembedding matrix, i.e. LogitLens (nostalgebraist, 2020), the trained probes predominantly map to toxic tokens (Table 1).

Validating linear probes. To validate that these probes represent toxicity, we apply *activation steering* (Zou et al., 2025; Panickssery et al., 2024) by subtracting a scaled probe W_{Toxic} from the final-layer residual stream $\mathbf{x}^{L-1}$ at each token position:

$$\mathbf{x}^{L-1}_{\text{steered}} = \mathbf{x}^{L-1} - \alpha W_{\text{Toxic}},$$

where α is selected to preserve language quality (perplexity and F1) of pre-trained models (see Appendix Table 11). Increasing α further reduces toxicity scores but raises perplexity (sAppendix Table 12). Table 2 shows that probe-based steering consistently reduces toxicity scores, validating their effects in eliciting toxic outputs. We therefore include it as a baseline for toxicity reduction.

Table 1: *The four toxic probes predominantly project to toxic tokens in the vocabulary space.* Warning: these examples are highly offensive.

Model	**Top tokens projected by probes**
GPT-2-355M	f*ck, c*nt, a**hole, holes, d*ck, wh*re
Llama-3.1-8B	en, kommen, F*CK, iyah, f*ck, dirty
Gemma-2-2B	rungsseite, fu*k, Fu*king, SH*T, a**hole
Mistral-7B	sh*t, f*ck, assh, bullsh*t, f*cked, a**hole

Per-neuron toxicity change via projection. For per-neuron contributions, we track how the toxic representation changes at each MLP neuron during DPO via its *change in projection* onto the probe:

$$\Delta_{\text{Toxic},i} = (m_i^{\text{pre}}\mathbf{v}_i^{\text{pre}} - m_i^{\text{dpo}}\mathbf{v}_i^{\text{dpo}}) \cdot \frac{W_{\text{Toxic}}}{\|W_{\text{Toxic}}\|_2}, \quad (2)$$

where $m_i^{\text{pre}}\mathbf{v}_i^{\text{pre}}$ and $m_i^{\text{dpo}}\mathbf{v}_i^{\text{dpo}}$ are the activated components of the i-th value vector before and after DPO; the activation scores m_i^{pre} and m_i^{dpo} are averaged over 20 generated tokens for all prompts in RealToxicityPrompts. This approach, known as *direct feature attribution* (Makelov et al., 2024; Arditi et al., 2024), quantifies each neuron's contribution to writing to the toxicity representation.

3.3 Activation Patching

Throughout our work, we apply *activation patching* (Zhang and Nanda, 2024) as a counterfactual method to isolate the effect of specific neurons on toxicity scores. For a pre-trained model and a set of MLP value vectors, we set their activations to match its post-DPO counterpart, based on the mean activation of 1,199 RealToxicityPrompts and 20 generated tokens per prompt. We then measure the resulting change in the toxicity scores.

4 Toxic Neurons Are Not Enough

We start by revisiting the claims in Lee et al. (2024): (a) DPO reduces toxicity primarily by dampening the activation of toxic neurons, and (b) this arises from shifts in earlier layer weights. We show that

Table 2: *Toxicity (Toxic), log perplexity (PPL), and F1 scores with activation patching and editing.* Across models, patching toxic neurons—whether those with toxic tokens or the top 256—yields only a limited drop in toxicity scores than DPO (Section 4). In contrast, patching all four of our identified groups matches or outperforms DPO (Section 5.2). Our activation editing method can outperform DPO, probe-based steering and patching all four groups (Section 6). Green highlights the editing parameters that best compete with DPO while preserving F1 scores.

Type	**Intervention**	**GPT-2-355M**			**Llama-3.1-8B**			**Gemma-2-2B**			**Mistral-7B**		
		Toxic	**PPL**	**F1**	**Toxic**	**PPL**	**F1**	**Toxic**	**PPL**	**F1**	**Toxic**	**PPL**	**F1**
Baselines	None	0.545	3.08	0.193	0.496	1.94	0.225	0.488	4.61	0.231	0.507	1.76	0.221
	Steering with probe	0.310	3.19	0.191	0.335	2.72	0.187	0.260	5.52	0.228	0.350	2.23	0.220
	DPO	0.210	3.15	0.195	0.241	2.69	0.221	0.245	5.15	0.228	0.191	2.01	0.223
Activation patching (Sec 5.2)	Patch toxic neurons	0.479	3.09	0.193	0.491	1.94	0.225	0.487	4.61	0.231	0.505	1.76	0.232
	Patch 256 neurons	0.465	3.07	0.193	0.488	1.94	0.225	0.482	4.61	0.231	0.455	1.76	0.232
	Patch TP↓	0.407	3.07	0.191	0.488	1.94	0.223	0.470	4.87	0.235	0.502	1.80	0.229
	Patch TP↓+AN↓	0.216	3.08	0.183	0.465	1.94	0.221	0.337	4.59	0.224	0.307	1.76	0.227
	Patch TP↓+AN↓+TN↓	0.194	3.08	0.170	0.391	1.94	0.208	0.307	4.59	0.217	0.238	1.81	0.218
	Patch four groups	0.139	3.08	0.170	0.278	1.94	0.207	0.260	4.58	0.213	0.138	1.78	0.209
Activation editing (Sec 6, probe-based)	$\alpha = 0.01, \beta = 0.8$	0.123	3.08	0.179	0.045	2.19	0.186	0.199	4.54	0.188	0.038	1.77	0.179
	$\alpha = 0.01, \beta = 0.6$	0.159	3.08	0.181	0.183	2.11	0.193	0.200	4.56	0.201	0.098	1.77	0.196
	$\alpha = \mathbf{0.01}, \beta = \mathbf{0.55}$	0.203	3.08	0.183	0.241	1.96	0.196	0.216	4.56	0.210	0.125	1.77	0.202
Activation editing (Sec 6, probe-free)	$\alpha = 0.01, \beta = 0.8$	0.139	3.08	0.176	0.116	5.82	0.200	0.218	4.54	0.180	0.057	1.77	0.191
	$\alpha = \mathbf{0.01}, \beta = \mathbf{0.6}$	0.238	3.08	0.178	0.258	2.28	0.210	0.216	4.57	0.203	0.162	1.77	0.200
	$\alpha = 0.01, \beta = 0.55$	0.282	3.08	0.180	0.318	2.24	0.204	0.250	4.58	0.198	0.239	1.77	0.201

(a) only partially explains the drop in toxicity, and in Section 5, we show that the weight shifts (b) are more nuanced than simply bypassing toxic neurons.

First, we directly measure the effect of dampening toxic neurons. We define toxic neurons by adapting the method of Lee et al. (2024): we identify the top N (= 256)[1] MLP value vectors with the highest cosine similarity to the toxic probe W_{Toxic}. In a second variant, we identify a smaller subset of interpretable value vectors. To do so, we unembed each value vector and consider it as toxic if any of its top-10 nearest tokens are toxic. We adopt LLM-as-a-judge (Zheng et al., 2023) using GPT-4o (OpenAI, 2024) to evaluate whether a token is considered toxic (e.g. curse words, slurs, sexual content). See Appendix Table 14 for the tokens projected by these toxic value vectors.

We then counterfactually isolate their effect on toxicity scores using activation patching (Section 3.3). Namely, for a pre-trained model, we set the activations of toxic value vectors to that of its post-DPO counterpart.

Table 3 reports the number of toxic neurons per model and the percentage reduction in toxicity scores through patching. Toxic neurons comprise fewer than 0.05% of all MLP neurons, yet account for as little as 2.5% to 24% of the reduction in toxicity scores, depending on the model. As patching captures interactions between toxic and non-toxic neurons, these results suggest that toxic neurons only account for a small portion of DPO's effect, rendering Lee et al. (2024)'s claim that DPO primarily dampens toxic neurons as incomplete.

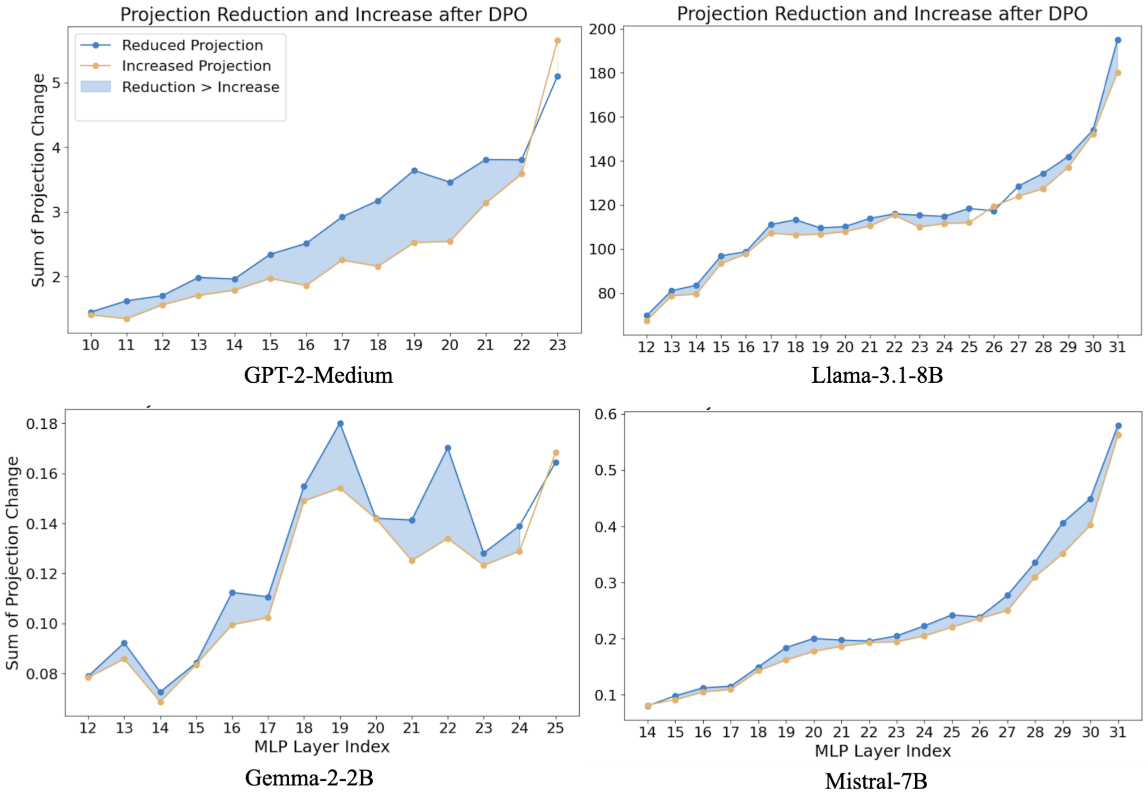
Table 3: *The number of toxic neurons per model and percentage decrease in toxicity scores after patching them.* The first row reports the number of toxic neurons unembed to toxic tokens. The second row reports results for the top 256 toxic-aligned neurons. The percentage decrease is the proportion of toxicity score reduction from patching toxic neurons, relative to the total reduction by DPO (see Table 2 for full scores).

GPT-2 355M	**Llama 3.1-8B**	**Gemma 2-2B**	**Mistral 7B**
59 (19.7%↓)	7 (1.96%↓)	3 (0.41%↓)	14 (0.63%↓)
256 (23.9%↓)	256 (3.14%↓)	256 (2.47%↓)	256 (16.5%↓)

5 A Deeper Look at DPO Weight Shifts

Next, we show that the weight shifts from DPO are more nuanced than simply bypassing toxic neurons.

[1] This number is based on Lee et al. (2024)'s number (128). We double the number of accommodate larger model sizes, but see similar results with the original 128 vectors.



GPT-2-Medium

Llama-3.1-8B

Gemma-2-2B

Mistral-7B

Figure 1: *DPO balances opposing toxicity writing across MLP layers.* Blue dots show total projection reduction per layer, orange dots show the total increase, both after DPO. The shaded blue areas illustrate how these opposing effects cancel out and lead to a net toxicity reduction. Projection changes grow with layers when measured against last-layer probe. Net changes in first ≈ 10 layers are negligible and omitted; see Appendix Table 5 for the full graph.

5.1 DPO Balances Opposing Effects

Across all models, DPO makes minimal adjustments to the MLP weights. All MLP value vectors have a cosine similarity of 0.99 before and after DPO, likely due to KL divergence regularisation (Rafailov et al., 2024). However, these small weight changes ($\mathbf{v}_i^{\text{pre}} \approx \mathbf{v}_i^{\text{dpo}}$) accumulate and induce distributed activation shifts ($m_i^{\text{pre}} - m_i^{\text{dpo}}$) across **all** MLP neurons. Most neurons undergo average shifts ranging from 0.66% (Llama-3.1-8B) to 16.71% (Mistral-7B), and substantial variation across neurons (see Appendix Figure 4).

These distributed activation shifts lead approximately half of all neurons (52%∼58% across models) reducing their projection onto the toxic direction ($\Delta_{\text{Toxic},i} > 0$) and the other half increasing it ($\Delta_{\text{Toxic},i} < 0$) (see Appendix Table 18). Figure 1 illustrates how these opposing neuron effects accumulate and balance out at each MLP layer, creating a net toxicity reduction. This suggests that DPO does not simply suppress toxic signals, but rather delicately redistributes them, balancing a trade-off across all MLP neurons.

5.2 Four Neuron Groups Reduce Toxicity

Building on this, we study value vectors that **reduce** toxic projections ($\Delta_{\text{Toxic},i} > 0$), as they likely contribute to toxicity reduction during DPO. We categorise them into four mutually exclusive groups, and study their collective effect.

Table 4 defines the four neuron groups, categorised by their alignment with the toxicity direction (**T**oxic-aligned vs. **A**nti-toxic-aligned) and their pre-DPO activations (**P**ositive vs. **N**egative). TP ↓, TN ↓ have positive alignment with toxicity, while AP ↓, AN ↓ have negative alignment. All groups reduce toxicity projection during DPO (↓). Table 5 shows the proportions of neurons in each group across models. Note that Lee et al. (2024) only considers the neurons in TP ↓.

Figure 2c visualises how the four groups reduce toxicity writing via activation shifts in Llama-3.1-8B, with similar patterns seen in all models (see Appendix Figure 6). The activations of each group are shifted in accordance to their orientation with respect to the toxic probe. Namely, toxic-aligned weights (TP ↓, TN ↓) drop in activations while

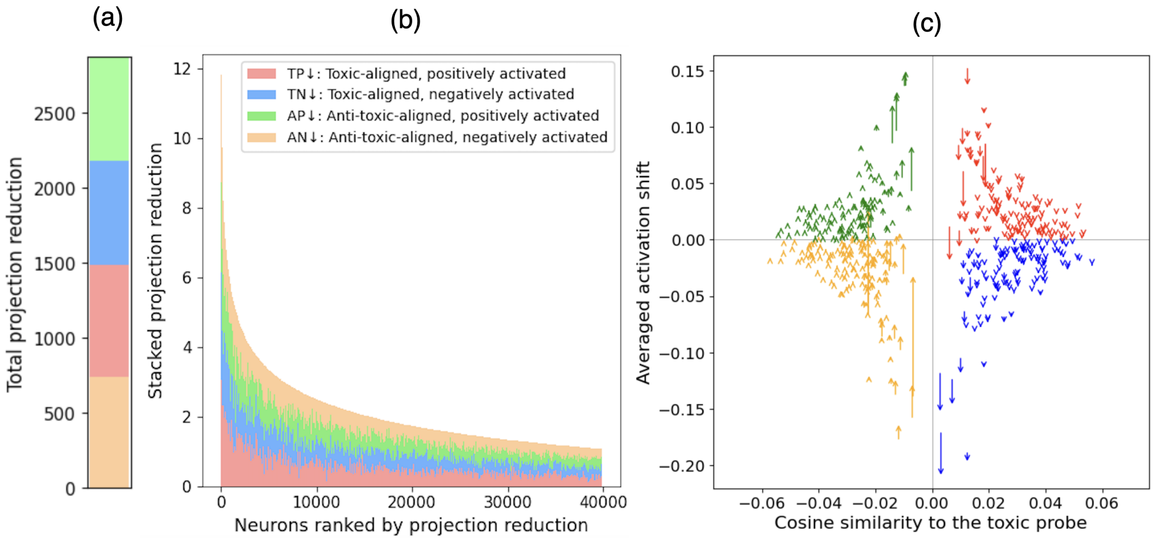


Figure 2: *Four neuron groups collectively reduce toxicity during DPO, shown for Llama-3.1-8B.* The same four groups emerge consistently across models, while panels (a) and (b) show differing patterns for the other three models (see Appendix Figure 6). (a) Proportion of toxicity reduction per group, showing balanced contributions; (b) Cumulative toxicity reduction for top 40,000 neurons (ranked by projection reduction), with groups showing similar reduction rates; (c) Per-group activation shifts during DPO for the top 2,000–2,500 neurons, where each group shifts according to their orientation relative to the toxic probe.

anti-toxic aligned weights (AN ↓, AP ↓) see an increase in activations (promotion of "anti-toxicity").

Table 4: *Definitions of four neuron groups reducing toxicity projections* ($\Delta_{\text{Toxic, i}} > 0$). *Alignment with probe* (T vs. A) indicates whether the neuron's value vector $\mathbf{v}$ aligns positively or negatively with the toxic probe W_{Toxic} ($\mathbf{v} \cdot W_{\text{Toxic}} > 0$ or $\mathbf{v} \cdot W_{\text{Toxic}} < 0$).

Group	Alignment with probe	Pre-DPO activation	Projection change
TP ↓	Toxic-aligned	Positive	Reduced (↓)
TN ↓	Toxic-aligned	Negative	Reduced (↓)
AP ↓	Anti-toxic-aligned	Positive	Reduced (↓)
AN ↓	Anti-toxic-aligned	Negative	Reduced (↓)

Table 5: *Proportions of four-neuron-group among all neurons reducing toxicity projection (↓).* Proportions are more balanced across larger LLMs. The *Sum* column shows the total number of neurons per model.

Model	TP ↓	TN ↓	AP ↓	AN ↓	Sum
GPT-2-355M	6.9%	39.1%	3.2%	50.9%	57,501
Llama-3.1-8B	25.4%	24.4%	24.6%	25.5%	239,460
Gemma-2-2B	28.8%	21.3%	21.3%	28.6%	123,898
Mistral-7B	29.7%	20.3%	20.2%	29.8%	238,236

Anti-toxic value vectors. What do "anti-toxic" value vectors encode? Geometrically, some anti-toxic value vectors essentially lie at the antipode of toxic semantic clusters. Namely, we take value vectors with high cosine similarity scores to $-1 \times W_{\text{Toxic}}$ (i.e. anti-toxic value vectors). We then multiply these value vectors by -1, unembed them, and inspect their nearest neighbors. Table 6 show examples of toxic tokens they project to (see Appendix Table 15 for more). To summarise, DPO also promotes anti-toxicity by increasing the activation of anti-toxic AN ↓, AP ↓ neurons.

Table 6: *Examples of anti-toxic value vectors (with reversed signs) that project to toxic tokens in Logit Lens.* Warning: these examples are highly offensive.

Model	Vector	Top tokens
GPT2	$-1\times\mathbf{v}_{11}^{1307}$	d*mn, darn, kidding, freaking, piss
Llama3	$-1\times\mathbf{v}_{25}^{14671}$	f*ck, f*cked, f*cking, sh*t, F*CK
Gemma2	$-1\times\mathbf{v}_{14}^{7822}$	f*cking, godd*mn, f*ck, sh*t
Mistral	$-1\times\mathbf{v}_{14}^{14693}$	sh*t, f*ck, Block, piss, f*cking

Why negatively activated? Negatively activated neurons (including TN ↓, AN ↓) take a large portion of MLP neurons—approximately 50% in three larger models and 87% in GPT-2 Medium (see Appendix Table 13). This results from the activation functions used in modern LLMs: GeLU (GPT-2), GeLU-Tanh (Gemma), and SiLU (Llama, Mistral), which allow neurons to retain small negative activations for negative inputs (Hendrycks and Gimpel, 2023). This allows plenty of neurons to remain weakly active and contribute marginally to the toxicity representation through their activation shifts.

Four groups reduce toxicity at different rates. When ranking neurons by their reduction of toxicity projection, the four groups show different patterns. In Llama-3.1-8B, all groups contribute evenly, maintaining balanced shares of top-ranked neurons (Figure 2b). In contrast, the other three models show TP ↓ dominating among top-ranked neurons, while AN ↓ gradually gains influence in later ranks—a trend most evident in GPT-2-Medium (see Appendix Figure 6). As a result, TP ↓ and AN ↓ dominate the overall toxicity reduction.

Reduction peaks at later layers. We observe an overall increasing trend in toxicity reduction across MLP layers for all neuron groups (see Appendix Figure 8). This suggests that the four groups collectively steer each layer away from toxicity, with later layers showing the strongest suppression of toxic outputs. This upward trend may be partly due to the probes being extracted from the final layer.

Activation patching confirms the collective effects of four groups. Finally, we confirm the collective effect of the four groups with activation patching. This post-hoc analysis assumes that we know the activations of each group after DPO and analyses their effects counterfactually. Namely, we patch the activations of each neuron group, one group at a time, in the pre-trained model to match that of the post-DPO model.

Table 2 shows that sequentially patching each neuron group further reduces toxicity scores across all models. This confirms the contributions of both anti-toxic and negatively activated groups to DPO's effects. Across models, patching all the four groups either surpasses or closely matches DPO's toxicity reduction, and consistently outperforms probe-based steering. It also has minimal impact on perplexity and only slightly reduces F1 scores. This activation patching outperforms DPO likely because we do not patch neurons that increase toxicity projection after DPO (Section 5.1). As a sanity check, patching all neurons that increase toxicity projection (↑) during DPO leads to higher toxicity scores across models, consistent with their projection changes (see Appendix Table 19).

6 Activating Editing to Replicate DPO

Based on our insights, we demonstrate two simple methods to replicate DPO's effects by directing editing activations. These methods only rely on a toxicity representation (e.g. a probe) and do not require any weight updates nor a pairwise preference dataset, which is not always readily available. Unlike the previous activation patching analyses, here we do not assume access to post-DPO activations.

Probe-based activation editing. Previously, we focused on neuron groups had a reduction in toxicity projections (i.e., $\Delta_{\text{Toxic, i}} > 0$) (Section 5.2). However, knowing whether a neuron undergoes a increase or decrease in toxicity projection requires access to post-DPO activations (see Equation 2). To remove this dependency, here we re-categorise the neuron groups based solely on their alignment with the toxicity probe and their pre-DPO activations, and do not consider their projection changes (hence notated as TP as opposed to TP ↓).

Given our new neuron groups (TP, TN, AP, AN), we leverage two key insights learned from DPO: activation shifts are distributed across all neurons (Section 5.1), and the direction of activation shifts for toxicity reduction depends on the orientation of the value vector (Section 5.2, Figure 2c).

Follow these insights, we sample a fraction β (%) of neurons from each group and minimally adjust their activations. For toxicity-aligned groups (TP, TN), we slightly decrease their activations by a factor of α (%), while for anti-toxicity-aligned groups (AP, AN) we slightly increase them. As TN and AN have negative activations, we flip the sign of α accordingly:

$$\begin{aligned} m^{\text{edit}}_{\text{TP}_\beta} &= (1-\alpha) m^{\text{pre}}_{\text{TP}_\beta}; \quad & m^{\text{edit}}_{\text{TN}_\beta} &= (1+\alpha) m^{\text{pre}}_{\text{TN}_\beta} \\ m^{\text{edit}}_{\text{AP}_\beta} &= (1+\alpha) m^{\text{pre}}_{\text{AP}_\beta}; \quad & m^{\text{edit}}_{\text{AN}_\beta} &= (1-\alpha) m^{\text{pre}}_{\text{AN}_\beta} \end{aligned} \tag{438}$$

where TP_β, AN_β, TN_β, and AP_β denote the β-fraction of neurons in each group, and m^{pre} are their pre-trained activations. Again, here we do not rely on any post-DPO information (i.e., m^{DPO}).

Table 2 shows our results for selected hyperparameters α and β. These hyperparameters reflect our insights: a majority of neurons (high β value) undergoes small shifts (small α value). We find that selecting the top-β fraction of neurons ranked by cosine similarity with the toxicity probe is most effective in reducing toxicity scores. In particular, selecting $\beta = 55\%$ yields the best trade-off between toxicity reduction and F1 scores, consistent of our earlier finding that DPO reduces toxicity writing in roughly half of all neurons (Section 5.1). This approach outperforms both DPO and probe-based steering in toxicity reduction while preserving perplexity across pre-trained models, with only a slight decrease in F1 scores. Further increasing β (e.g., to 0.8) leads to greater toxicity reduction at the cost of F1 drops. Alternative sampling strategies for selecting the top-β neurons (e.g., based on ascending absolute activation values) yield similar results across models (see Appendix Table 19).

Probe-free activation editing. While the previous activation editing method does not require pairwise preference data, it still relies on a latent toxicity representation, for which we use our probe. While a probe does not require pairwise preference data, it still needs labelled classification data (Section 3).

Here, we demonstrate that activation editing can be performed even without a probe by leveraging an alternative toxicity representation. Namely, prior works have observed a close relationship between concept representations in the model's hidden layers and the token embedding space (Lee et al., 2025). Similarly, we observe that toxic tokens are nearest neighbors to our probes in the token embedding space (Table 1). Motivated by this, we replace the probe with a contrastive vector derived directly from token embeddings.

To construct this vector, we simply select sets of toxic and non-toxic token embeddings in each model and compute the difference between their mean embeddings (Table 7). This bypasses the need to train a probe model. We then apply the same activation editing method as described above.

Table 7: *Toxic and non-toxic tokens used to compute the contrastive vector.* The contrastive vector is derived by subtracting the mean embedding of non-toxic tokens from that of toxic tokens.

Toxic	fu*k	sh*t	cr*p	da*n	a**hole
Non-toxic	hello	thanks	friend	peace	welcome

The last rows of Table 2 show that this approach yields results comparable to our probe-based method. These results together validate our mechanistic understanding of DPO and offer a proof-of-concept alternative when weight updates are costly or training data is not readily available.

7 Discussion and Conclusion

Our work provides a mechanistic understanding of how DPO reduces toxicity across four LLMs. Using activation patching, we show that prior explanations are incomplete: a small number of toxic neurons associated with toxic tokens (Lee et al., 2024) cannot fully explain DPO's effects. This explanation also relies on a monosemantic view of neurons, an assumption disputed by prior work (Elhage et al., 2022). Instead, DPO induces distributed activation shifts across all MLP neurons to produce a net toxicity reduction.

To characterise these distributed effects, we identify four neuron groups that play distinct roles in toxicity reduction and show that their combined effect replicates the effect of DPO.

Building on these insights, we develop an activation editing method mimicking DPO by applying distributed activation shifts along a learned toxicity representation. We explore two options for this representation: a probe model and a contrastive vector derived from token embeddings. This method outperforms DPO in reducing toxicity while preserving perplexity, all without any weight updates.

In summary, our work provides a more complete understanding of how DPO reduces toxicity and introduces a efficient, training-free alternative.

The shallowness of safety. DPO's tendency to spread activation shifts thinly across the network suggests that pre-trained harmful capabilities are not removed, but merely masked. As a result, small disruptions anywhere in the model, not just in toxic neurons, can potentially breach the safety barrier and reactivate harm. This extends prior findings on the shallowness of safety fine-tuning from the activation perspective (Jain et al., 2024; Qi et al., 2024). These distributed shifts likely arise as a by-product of regularisation to preserve pre-training performance, hinting at a deeper trade-off: the shallow safety may be an inherent cost of maintaining language quality. This diluted effect is further compounded by the use of smooth activation functions (dicussed in Section 5.2), which allow many weakly active neurons to marginally participate in toxicity writing. As a result, much of the model's capacity for toxicity reduction remains untapped—we observe many MLP neurons actually increase their toxicity projection during DPO (Section 5.1). In contrast, our activation editing method offers a more targeted alternative by explicitly steering activations toward reducing toxicity. This may explain why it achieves greater toxicity reduction than DPO, despite applying smaller average activation changes. Taken together, our findings point to the value of exploring more interpretable safety interventions as a path beyond shallow tuning.

Limitations

Projection to a toxic subspace. In this work, we use a linear probe to capture an aggregated toxicity representation, following common practice in the literature (Ferrando et al., 2024; Ravfogel et al., 2022). However, it may be possible that toxicity manifest along multiple directions, each capturing different aspects such as hate speech or abusive language, thus better represented as a subspace (Uppaal et al., 2024). We thus conduct an initial analysis on GPT-2-Medium. We construct the toxic subspace by applying Singular Value Decomposition (SVD) to the top 128 toxic-aligned value vectors and selecting the top singular directions, each of which projects to different toxic tokens (see Appendix G). However, we find that most value vectors show inconsistent alignment across the three directions and mixed projection changes after DPO. A single value vector can be "toxic-aligned" in one SVD direction and "anti-toxic-aligned" in another, also reducing toxicity along one axis while increasing it in another. Such inconsistencies make it difficult to assign neurons to coherent neuron groups as in our approach. We therefore leave a more robust analysis of toxic subspace projections to future work.

Assumptions for the projection. We use projection to estimate each neuron's contribution to toxicity (Equation 2), assuming that neurons contribute proportionally along their activated directions. However, toxicity representations may be distributed across more complex linear combinations of neurons. Alternative tools, such as sparse autoencoders (SAEs) (Bricken et al., 2023; Cunningham et al., 2023), which learn linear feature compositions through autoencoder reconstruction, may offer a complementary perspective for tracing toxic feature changes back to specific neurons.

Generalise the four neuron groups across tasks and models. DPO is inherently a binary algorithm, as it is trained on pairwise preference data. The four neuron groups we identify naturally reflect this binary structure, with activations shifting along the representation of a binary concept. Accordingly, we expect similar group structures to emerge in other binary safety-related tasks beyond toxicity (e.g., biased vs. unbiased content, factual vs. misinformation) under DPO—a direction we leave for future work.

These four neuron groups may also persist in general instruction-tuned models (e.g., those trained with supervised fine-tuning or RLHF) on binary tasks, likely operating through distributed activation shifts due to regularisation. We leave this as another direction for exploration.

Generalising the activation editing method to more tasks. Our activation editing method requires only a linear concept representation, which can be derived from a probe or token embeddings—both relatively cheap to obtain. Future work could extend our method to other safety-related tasks (e.g., bias or misinformation) where such representations are available, or to general tasks where the target behavior can be captured by representative tokens (e.g., sentiment polarity, political stance).

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *Preprint*, arXiv:2406.11717.

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

cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, nithum, and Will Cukierski. 2017. Toxic comment classification challenge. Accessed: 18-May-2025.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *Preprint*, arXiv:2309.08600.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, and Tom Henighan et al. 2022. Toy models of superposition. *Preprint*, arXiv:2209.10652.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R. Costa-jussà. 2024. A primer on the inner workings of transformer-based language models. *Preprint*, arXiv:2405.00208.

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. *Preprint*, arXiv:2309.00770.

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

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

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. *Preprint*, arXiv:2012.14913.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and Abhishek Kadian et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Laura Hanu. 2020. Detoxify. `https://github.com/unitaryai/detoxify`. Accessed: 18-May-2025.

Dan Hendrycks and Kevin Gimpel. 2023. Gaussian error linear units (gelus). *Preprint*, arXiv:1606.08415.

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

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, and et al. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

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

Andrew Lee, Melanie Weber, Fernanda Viégas, and Martin Wattenberg. 2025. Shared global and local geometry of language model embeddings. *Preprint*, arXiv:2503.21073.

Aleksandar Makelov, George Lange, and Neel Nanda. 2024. Towards principled evaluations of sparse autoencoders for interpretability and control. *Preprint*, arXiv:2405.08366.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

nostalgebraist. 2020. Interpreting GPT: The logit lens. *AI Alignment Forum*. Accessed: 18-May-2025.

OpenAI. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2024. Steering llama 2 via contrastive activation addition. *Preprint*, arXiv:2312.06681.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. 2024. Safety alignment should be made more than just a few tokens deep. *Preprint*, arXiv:2406.05946.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2023. Fine-tuning aligned language models compromises safety, even when users do not intend to! *Preprint*, arXiv:2310.03693.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *Open AI*.

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

Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell. 2022. Linear adversarial concept erasure. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18400–18421. PMLR.

Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, and Cassidy Hardin et al. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Patrick Schober, Christa Boer, and Lothar A. Schwarte. 2018. Correlation coefficients: Appropriate use and interpretation. *Anesthesia & Analgesia*, 126(5):1763–1768.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Noam Shazeer. 2020. Glu variants improve transformer. *Preprint*, arXiv:2002.05202.

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

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does llm safety training fail? *Preprint*, arXiv:2307.02483.

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

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. 2023. Shadow alignment: The ease of subverting safely-aligned language models. *Preprint*, arXiv:2310.02949.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods. *Preprint*, arXiv:2309.16042.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, and et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, and et al. 2025. Representation engineering: A top-down approach to ai transparency. *Preprint*, arXiv:2310.01405.

Table of Contents

A Gated Linear Units 12

B MLP layer specification 12

C DPO training hyperparameters 12

D More results on toxic probes 13

E Negatively activated value vectors 13

F Logit lens tokens for value vectors 14

G Projecting value vectors to a toxic subspace 14

H More results on activation shifts 15

I More results on opposing neuron effects 15

J More results on four neuron groups 15

K More results on activation editing 16

A Gated Linear Units

In this section, we introduce Gated Linear Units (GLUs), which replace standard MLPs (Section 2) in recent models such as Llama, Gemma, Mistral (Shazeer, 2020).

GLUs introduce a gating mechanism that selectively controls information flow by computing the element-wise product of two linear projections, one of which is passed through a non-linearity σ:

$$\mathrm{GLU}^{\ell}(\mathbf{x}^{\ell}) = \left(\sigma(W_1^{\ell}\mathbf{x}^{\ell}) \odot W_2^{\ell}\mathbf{x}^{\ell}\right) W_V^{\ell},$$

where $W_1^{\ell}, W_2^{\ell}, W_V^{\ell} \in \mathbb{R}^{d_{mlp} \times d}$. The term $\sigma(W_1^{\ell}\mathbf{x}^{\ell})$ acts as the *gates*, blocking $W_2^{\ell}\mathbf{x}^{\ell}$ from propagating when the non-linearity (σ) is inactive.

We can still express GLUs as (see Equation 1):

$$\mathrm{MLP}^{\ell}(\mathbf{x}^{\ell}) = \sum_{i=1}^{d_{\text{mlp}}} m_i^{\ell} \mathbf{v}_i^{\ell},$$

where

$$m_i^{\ell} = \sigma(\mathbf{k}_i^{\ell} \cdot \mathbf{x}^{\ell}) \cdot (\mathbf{w}_i^{\ell} \cdot \mathbf{x}^{\ell}),$$

$\mathbf{k}_i^{\ell} \in \mathbb{R}^d$ and $\mathbf{w}_i^{\ell} \in \mathbb{R}^d$ are the i-th rows of W_1^{ℓ} and W_2^{ℓ}, respectively. For each MLP neuron i, $\mathbf{v}_i^{\ell}$ (rows of W_V^{ℓ}) is its *value vector* (Geva et al., 2021), and the scalar $m_i^{\ell} \in \mathbb{R}$ is an *activation score* that controls the scaling of the value vector $\mathbf{v}_i^{\ell}$.

This shows that, despite despite architectural differences in GLUs, our formulation in Equation 1 still holds, as it consists of value vectors scaled by a non-linear activation.

B MLP layer specification

In this section, we provide the MLP layer specifications for each model (Section 3.1).

Table 8 reports, for each model, the number of MLP layers, MLP hidden dimensions, activation function, and whether a gating mechanism is used.

Table 8: *MLP specifications for each model.* l is the number of MLP Layers, d is the residual stream dimension, d_{mlp} is the dimension of MLP hidden layer, σ is the activation function, *Gated?* indicates whether the model uses gated MLPs.

Model	l	d	d_{mlp}	σ	*Gated?*
GPT-2-355M	24	1024	4096	GeLU	×
Llama-3.1-8B	32	4096	14336	SiLU	✓
Gemma-2-2B	26	2304	9216	GeLUTanh	✓
Mistral-7B	32	4096	14336	SiLU	✓

C DPO training hyperparameters

In this section, we provide the hyperparameters for DPO training (Section 3.1).

Table 9 reports the shared hyperparameters across models. Table 10 reports the KL regularisation weight λ tuned in DPO to maintain pre-trained model's perplexity and F1 scores for each model.

Table 9: *Shared hyperparameters for DPO Training.*

Hyperparameter	**Value / Description**
Beta (β)	0.1 (preference strength)
Optimizer	RMSprop
Learning rate	1×10^{-5}
Warmup steps	150
Gradient accumulation steps	4
Batch size	4 (per step)
Evaluation batch size	8
Max input length	256 tokens
Max new tokens	64 tokens
Max prompt length	64 tokens
Epochs	5
Gradient clipping	Max norm = 10.0
Patience for early stopping	30 validations

Table 10: *The KL regularisation weight λ for each model.* λ is selected to maintain perplexity and F1 scores to pre-trained models.

Model	**KL weight (λ)**
GPT-2-355M	0.02
Llama-3.1-8B	0.1
Gemma-2-2B	0.05
Mistral-7B	0.05

D More results on toxic probes

In this section, we provide more results on validating toxic linear probes (Section 3.2).

Table 11 reports the test accuracies of linear probes on the Jigsaw Toxic Comment Classification dataset (90–10 split) (cjadams et al., 2017), with all probes achieving over 91% accuracy. It also reports the selected α values for probe-based steering that best preserve the pre-trained models' perplexity and F1 scores.

Table 11: *Validation accuracy of toxicity probes and scaling values α for probe-based steering.* α is selected to preserve the pre-trained perplexity and F1 scores.

Model	**Validation Accuracy**	α
GPT-2-355M	95.6%	30
Llama-3.1-8B	92.6%	2
Gemma-2-2B	96.1%	3
Mistral-7B	91.0%	5

Table 12 shows that in probe-based activation steering, increasing α beyond the selected values further reduces toxicity, but also increases perplexity and lowers F1 scores. This demonstrates a trade-off in steering: stronger steering reduces toxicity at the cost of general language quality.

E Negatively activated value vectors

In this section, we show that a large proportion of value vectors v_i are negatively activated by their activations m_i (Section 5.2).

Table 13 reports the percentage of MLP neurons that are negatively activated across models, showing that they constitute at least half of all MLP neurons.

Since GPT-2 Medium has a particularly high proportion of negatively activated neurons (over 87%), Figure 3 illustrates this by showing the average activations of the top 100 toxic-aligned neurons. Most of these value vectors remain negatively activated both before and after DPO, reflecting the impact of the GeLU activation function.

Table 12: *Toxicity (Toxic), log perplexity (logPPL), and F1 scores after probe-based steering with different α values.* Larger α reduces toxicity but increases perplexity and lowers F1 scores. *Bold* highlights the selected α values.

Model	**Method**	**Toxic**	**logPPL**	**F1**
GPT-2-355M	None	0.545	3.08	0.193
	Subtract (α=**30**)	0.310	3.19	0.191
	Subtract (α=40)	0.250	3.34	0.180
Llama-3.1-8B	None	0.496	1.94	0.225
	Subtract (α=**2**)	0.335	2.72	0.187
	Subtract (α=3)	0.267	3.53	0.180
Gemma-2-2B	None	0.488	4.61	0.231
	Subtract (α=**3**)	0.260	5.52	0.228
	Subtract (α=5)	0.251	5.64	0.226
Mistral-7B	None	0.507	1.76	0.231
	Subtract (α=**5**)	0.350	2.23	0.220
	Subtract (α=7)	0.319	2.63	0.212

Table 13: *Percentages of MLP neurons with negative pre-trained activations.* The three larger LLMs have approximately 50% of their MLP neurons negatively activated, whereas GPT-2 Medium has over 87%.

Model	**% neurons negatively activated**	**% neurons positively activated**
GPT-2-355M	87.28%	12.71%
Llama-3.1-8B	49.96%	50.04%
Gemma-2-2B	49.94%	50.06%
Mistral-7B	50.03%	49.97%

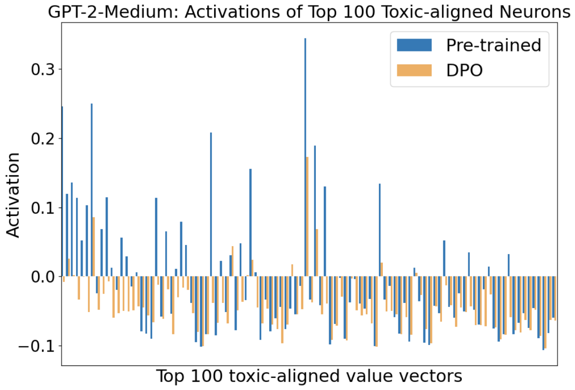


Figure 3: *Activations of the top 100 toxic-aligned neurons in GPT-2-Medium.* The activation m_i for each value vector is averaged over all prompts and 20 generated tokens. The majority of value vectors remain weakly negatively activated both before and after DPO.

F Logit lens tokens for value vectors

In this section, we present the Logit Lens tokens for selected value vectors.

Table 14 shows the top toxic-aligned value vectors that project to at least one toxic token among the top 10 nearest tokens (Section 4).

Table 15 shows example anti-toxic value vectors that, when sign-reversed, project to at least one toxic token across models (Section 5.2).

Table 14: *Examples of toxic value vectors that project to toxic tokens in LogitLens.* Value vectors are *ranked* by their cosine similarity with the toxicity probe W_{Toxic}. Warning: these examples are highly offensive.

GPT-2 medium

Vector	Rank	Top tokens
$\mathbf{W}_{\text{Toxic}}$	-	c*nt, f*ck, a**hole, d*ck, wh*re, holes
$\mathbf{v}_{19}^{770}$	1	sh*t, a**, cr*p, f*ck, c*nt, garbage
$\mathbf{v}_{13}^{253}$	18	c*m, d*ck, icles, icle, bo*bs, naughty
$\mathbf{v}_{7}^{3358}$	29	cr*p, whine, sh*t, uphem, shri, bullsh*t
$\mathbf{v}_{6}^{3972}$	50	f*cking, d*mn, sinful, hell, immoral

Llama-3.1-8B

Vector	Rank	Top tokens
$\mathbf{W}_{\text{Toxic}}$	-	en, kommen, F*CK, iyah, f*ck, dirty
$\mathbf{v}_{25}^{14065}$	1	sh*t, cr*p, rubbish, garbage, bullsh*t
$\mathbf{v}_{27}^{3863}$	7	waste, trash, garbage, junk, wastes
$\mathbf{v}_{18}^{13797}$	10	Shoot, oh, sh*t, cr*p, swore, allocator
$\mathbf{v}_{19}^{6041}$	18	d*mn, sh*t, DAM, dam, fig, D*mn

Gemma-2-2B

Vector	Rank	Top tokens
$\mathbf{W}_{\text{Toxic}}$	-	rungsseite, fu*k, Fu*king, SH*T
$\mathbf{v}_{4}^{2288}$	1	idiot, bastard, a*hole, fu*ks, moron
$\mathbf{v}_{18}^{2597}$	10	yes, ridiculous, absurd, ludicrous
$\mathbf{v}_{3}^{4727}$	22	sh*t, Sh*t, cr*p, sh*ts, sh*te, sh*tty

Mistral-7B

Vector	Rank	Top tokens
$\mathbf{W}_{\text{Toxic}}$	-	sh*t, f*ck, assh, bullsh*t, a**hole
$\mathbf{v}_{22}^{1061}$	1	fu*k, sh*t, bullsh*t, a**hole, sh*tty
$\mathbf{v}_{15}^{2454}$	4	fuck*ng, bullsh*t, stupid, sh*t, cr*p
$\mathbf{v}_{14}^{11281}$	34	sexual, sex, girls, women, dating, porn
$\mathbf{v}_{19}^{4689}$	45	cr*p, sh*t, d*mn, hell, b*tch, piss

Table 15: Examples of anti-toxic value vectors that, when sign-reversed, project to toxic tokens in Logit Lens. *Rank* gives the cosine similarity rank with $-1 \times W_{\text{Toxic}}$, reflecting how "anti-toxic" a neuron is. Warning: these examples are highly offensive.

GPT-2 medium

Vector	Rank	Top tokens
$-1 \times \mathbf{v}_{10}^{1882}$	1	maniac, ueless, thug, arrog, f*cking
$-1 \times \mathbf{v}_{11}^{1307}$	3	d*mn, darn, kidding, freaking, piss
$-1 \times \mathbf{v}_{15}^{301}$	4	harmful, worse, unfavorable, disturbing

Llama-3.1-8B

Vector	Rank	Top tokens
$-1 \times \mathbf{v}_{25}^{14671}$	2	f*ck, f*cked, f*cking, sh*t, F*CK
$-1 \times \mathbf{v}_{14}^{4997}$	19	s*cks, s*ck, adla, BackPressed, teri

Gemma-2-2B

Vector	Rank	Top tokens
$-1 \times \mathbf{v}_{14}^{7822}$	1	f*cking, godd*mn, f*ck, sh*t, d*mn
$-1 \times \mathbf{v}_{6}^{7099}$	2	f*cking, f*ck, f*cker, p*ss, F*ck
$-1 \times \mathbf{v}_{17}^{8418}$	13	idiot, idiots, stupid, moron, dumbass

Mistral-7B

Vector	Rank	Top tokens
$-1 \times \mathbf{v}_{14}^{14693}$	1	sh*t, f*ck, Block, piss, f*cking, bitch
$-1 \times \mathbf{v}_{14}^{8200}$	16	cr*p, nonsense, stupid, d*mn, ridiculous
$-1 \times \mathbf{v}_{17}^{14302}$	25	hell, d*mn, d*mned, f*ck, cr*p, sh*t
$-1 \times \mathbf{v}_{12}^{8139}$	36	f*cked, sh*t, bitch, sex, sexual, rape

G Projecting value vectors to a toxic subspace

In this section, we present initial results using a toxic subspace to capture toxicity representations in GPT-2-Medium and to perform projections (discussed in *Limitations*). We explain why we do not adopt this approach for neuron analysis, as it complicates the identification of coherent neuron groups.

Specifically, on GPT-2-Medium, we apply singular value decomposition (SVD) to the value vectors of 128 toxic-aligned MLP neurons, using the top three components as basis directions to capture different aspects of toxicity. We choose $N = 128$ because it yields a stable toxic subspace—adding more value vectors does not significantly expand it. Table 16 shows that these SVD vectors unembed to different toxic tokens, including offensive curse words ($\text{SVD}_{\text{Toxic}}[0]$), mild insults ($\text{SVD}_{\text{Toxic}}[1]$), and sexualised terms ($\text{SVD}_{\text{Toxic}}[2]$).

Follow Section 5.2, we attempt to identify neuron groups based on their projection changes onto the toxicity subspace. One approach is to compute a weighted sum of the SVD vectors (scaled by their singular values) to form a single combined direction, then measure projections onto it. However, this provides little advantage over using a standard

Table 16: *Logit Lens tokens for the top three SVD vectors extracted from 128 toxic-aligned neurons in GPT-2 Medium.* Each SVD direction captures a different aspect of toxicity. Warning: these examples are highly offensive.

Model	Top Tokens
$\text{SVD}_{\text{Toxic}}[0]$	f*ck, assh*le, f*cking, d*ck, sh*t, sl*t
$\text{SVD}_{\text{Toxic}}[1]$	d*mned, cr*p, stupid, darn, Godd, idiots
$\text{SVD}_{\text{Toxic}}[2]$	sex, boobs, chicks, sexy, vagina, breasts

toxicity probe. Instead, we project each value vector onto each SVD vectors individually.

Since the SVD vectors are orthonormal, the total projection onto the toxic subspace is equivalent to summing the projections onto each SVD direction. Thus to identify neurons reducing toxicity, we compute each value vector's cosine similarity with the SVD vectors, along with their projections before and after DPO.

We find that 74.7% of value vectors have conflicting signs of alignment across the SVD directions—that is, they align positively with at least one vector and negatively with another. This complicates defining whether a neuron is "toxic-aligned". Similarly, 74.3% of neurons show inconsistent projection change after DPO, reducing toxicity along one direction while increasing it along another.

These inconsistencies make it impossible to identify coherent neuron groups that reduce toxicity across all SVD directions, i.e. across the toxic subspace. This also means that each SVD direction induces its own set of contradictory neuron groups. More importantly, this prevents us from linking toxicity scores to specific neuron groups via activation patching (Section 5.2), as a single neuron can simultaneously increase and decrease toxicity depending on the direction.

For these reasons, we choose not to proceed with subspace projection for neuron analysis and instead focus on the single-probe approach.

H More results on activation shifts

In this section, we provide more results on DPO-induced activation shifts by presenting their distributions and analyse whether they occur systematically with neuron properties. These results complement Section 5.1.

Figure 4 shows the distribution of activation shifts across models. Most neurons have small activation shifts around the mean but substantial variation in the tails.

Table 17 presents the results of a Pearson correlation analysis (Schober et al., 2018) between DPO-induced activation shifts and neuron properties. The analysis reveals no correlation between activation shifts and the "toxicity level" of a neuron—measured by its cosine similarity with the toxic probe—and only a weak positive correlation with pre-trained activations. While this may suggest a slight tendency for DPO to push activations toward zero, the pattern is likely due to a regression-to-the-mean effect, thus more of a statistical artifact than an intentional toxicity-reduction mechanism. These findings indicate that DPO-induced activation shifts are largely random.

I More results on opposing neuron effects

In this section, we provide more statistics and visualisations on the opposing neuron effects (Section 5.1).

Table 18 shows the percentage of neurons reducing toxicity projection ($\Delta_{\text{Toxic},i} < 0$, denoted as ↓), ranging from 52% in Gemma-2-2B to 58% in GPT-2-Medium. This shows that DPO's activation shifts cause roughly half of the MLP neurons to reduce toxicity projection, while the other half increase it, revealing a trade-off in toxicity reduction.

Figure 5 visualises the opposing effects across all MLP layers, complementing Figure 1 by including the first 10 layers that were omitted.

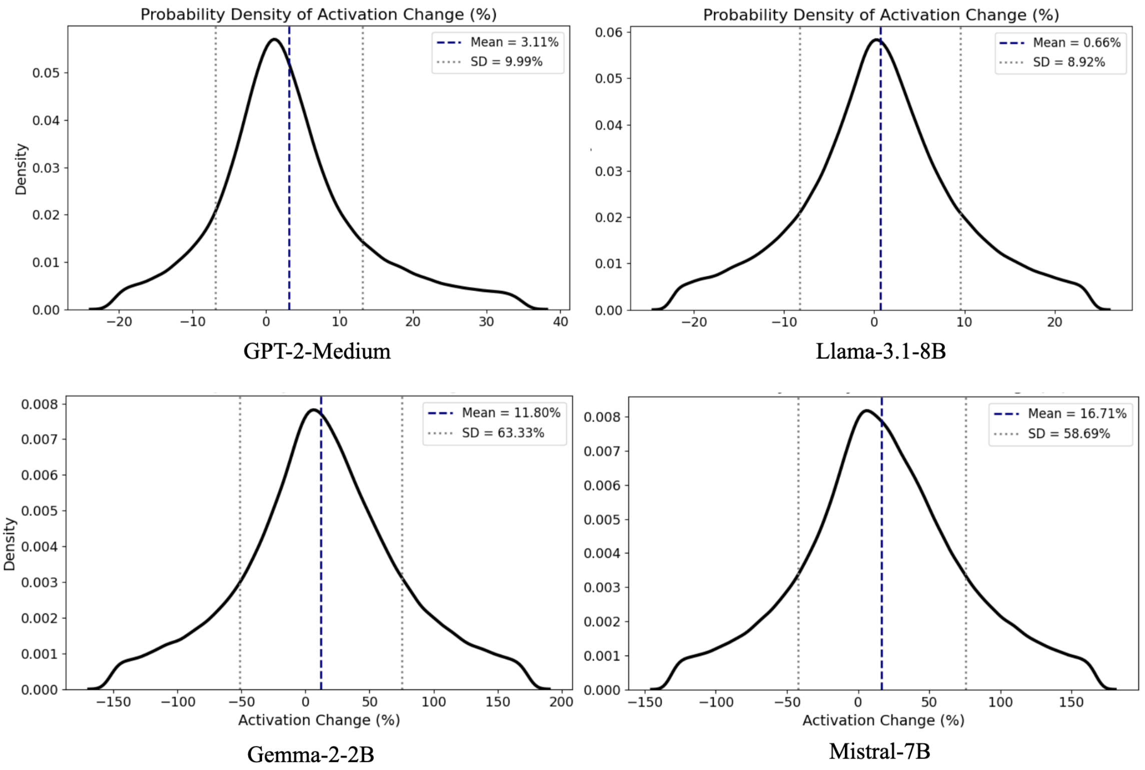
J More results on four neuron groups

In this section, we provide more visualisations on the four neuron groups (Section 5.2).

Figure 6 shows the four-group distributions for GPT-2-Medium, Gemma-2-2B, and Mistral-7B, repeating the analysis from Figure 2 for Llama-3.1-8B. In these three models, overall toxicity reduction is primarily driven by TP ↓ and AN ↓, which dominate the stacked bars in Figure 6a.

Figure 6b shows that the four groups reduce toxicity projection at different rates when neurons are ranked by their contribution. TP ↓ dominates among the top-ranked neurons, while AN ↓ becomes more prominent later, especially in GPT-2-Medium. Figure 7 further decodes this trend in GPT-2-Medium, where activation shifts become more evenly distributed in lower-ranked neurons.

Figure 6c demonstrates that each group shifts activations according to their orientation relative



GPT-2-Medium

Llama-3.1-8B

Gemma-2-2B

Mistral-7B

Figure 4: *Probability density of activation shifts ($m_i^{pre} - m_i^{dpo}$) during DPO.* Most neurons have small activation shifts around the mean, with more substantial variation in the tails. Gemma-2-2B and Mistral-7B show larger average shifts and standard deviations (SD) compared to the other two models.

Table 17: *Pearson correlation between activation shifts and neuron properties.* Activation shifts ($m_i^{\text{pre}} - m_i^{\text{dpo}}$) show no correlation with a neuron's "toxicity level" (measured by cosine similarity with the toxic probe), and only a weak positive correlation with pre-trained activations, which is likely a regression-to-the-mean effect.

Variables	**Metric**	**GPT-2-355M**	**Llama-3.1-8B**	**Gemma-2-2B**	**Mistral-7B**
Activation shift & probe alignment	Correlation p-value	0.004 0.252	0.001 0.487	0.004 0.071	0.003 0.045
Activation shift & pre-trained activation	Correlation p-value	0.263 **<0.0001**	0.033 **<0.0001**	0.098 **<0.0001**	0.347 **<0.0001**

Table 18: *Percentages of neurons reducing toxicity projection after DPO.* Across models, 52% to 58% of MLP neurons reduce their projection ($\Delta_{\text{Toxic},i} < 0$) onto the toxicity probe, while the remaining neurons increase it ($\Delta_{\text{Toxic},i} > 0$).

Model	**% neurons reduce projection (↓)**	**% neurons increase projection (↑)**
GPT-2-355M	58.49%	41.51%
Llama-3.1-8B	53.01%	46.99%
Gemma-2-2B	51.75%	48.25%
Mistral-7B	51.98%	48.02%

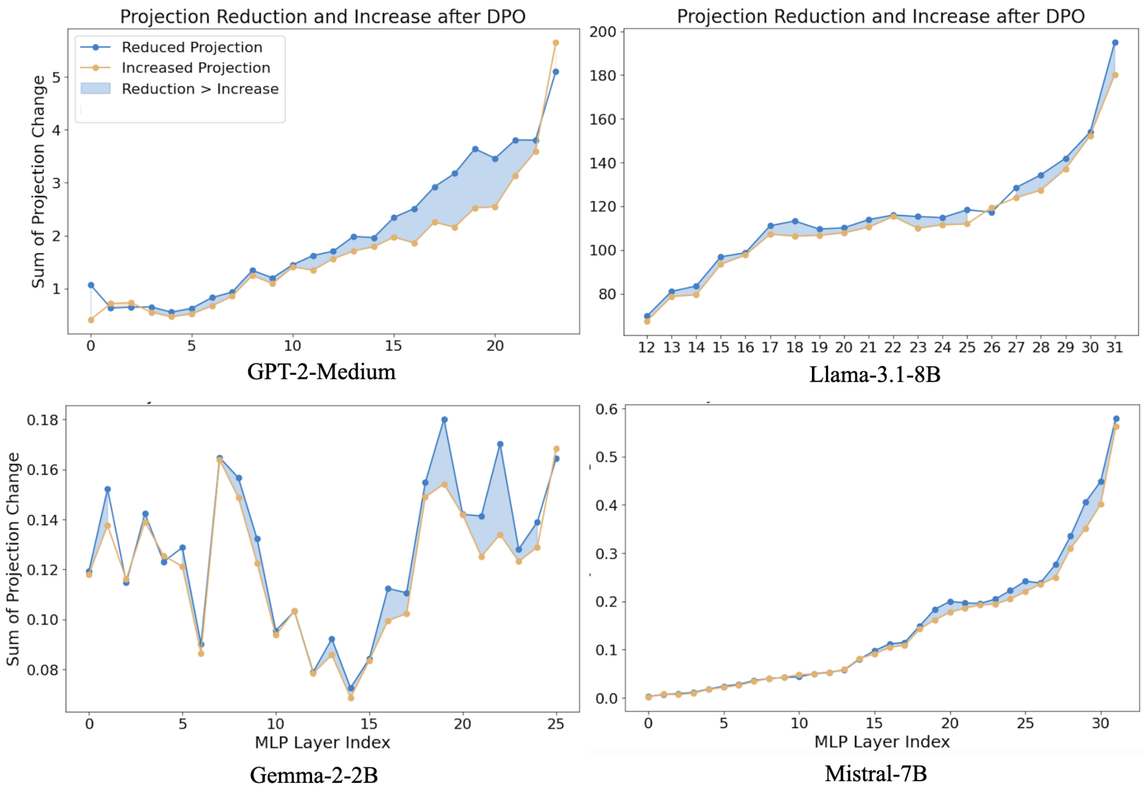
to the toxic probe, consistent with the pattern observed in Figure 2c.

Figure 8 shows toxicity reduction across layers for all four groups. The reduction generally increases through successive MLP layers, reflecting the cumulative effect of activation shifts, though this trend is less pronounced in Gemma-2-2B. These results suggest that layers progressively steer the residual stream away from toxicity, with later layers showing the strongest suppression of toxic outputs. The upward trend may be partly due to our use of final-layer probes for extraction.

K More results on activation editing

In this section, we present more results on activation editing (Section 6).

Table 19 extends our probe-based editing results, comparing two selection methods for the top-β



GPT-2-Medium

Llama-3.1-8B

Gemma-2-2B

Mistral-7B

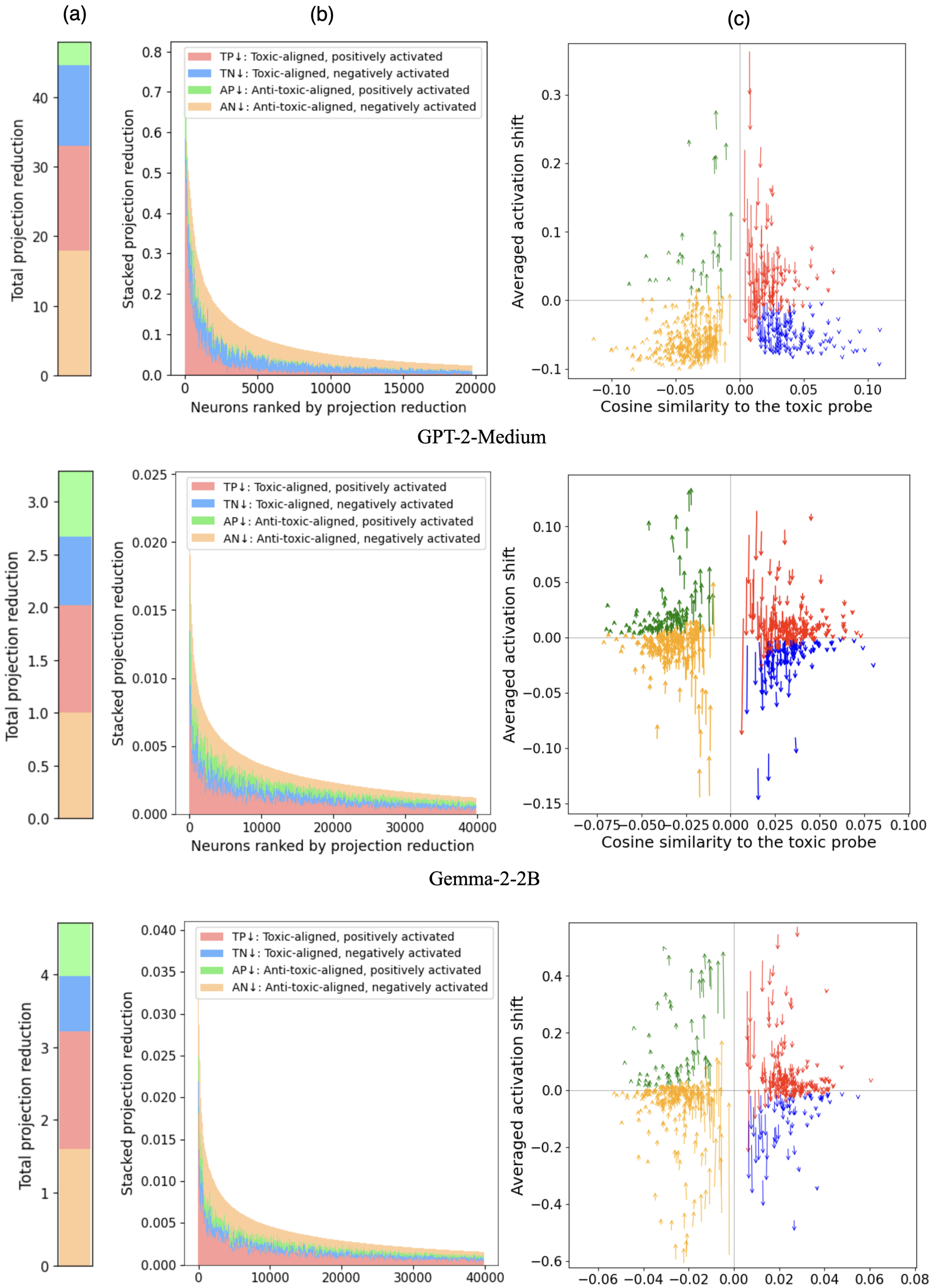
Figure 5: *DPO balances opposing toxicity writing across* ***all*** *MLP layers.* Blue dots show the total projection reduction per layer, while orange dots show the total increase, both after DPO. The shaded blue areas illustrate how the opposing effects cancel out and lead to a net toxicity reduction. Projection changes tend to grow in later layers when measured against the last-layer probe.

neurons: descending cosine similarity with probe (main results also in Table 2) and by ascending absolute activations. While both approaches work, the latter is slightly less effective and fails to surpass DPO for Gemma-2-2B.

As a sanity check, we also patching neurons with increased toxicity projection (↑) during DPO and find that they raise toxicity scores across models (Section 5.2).

Table 19: *Toxicity (Toxic), log perplexity (PPL), and F1 scores with activation patching and editing.* As a sanity check, patching neurons with increased toxicity projection (↑) raises toxicity scores. In probe-based editing, we compare two samping strategies for the top-β neurons: descending cosine similarity with the probe and ascending absolute activation values. For both approaches, Green highlights the editing parameters that best compete with DPO while preserving F1 scores.

Type	**Intervention**	**GPT-2-355M**			**Llama-3.1-8B**			**Gemma-2-2B**			**Mistral-7B**		
		Toxic	**PPL**	**F1**	**Toxic**	**PPL**	**F1**	**Toxic**	**PPL**	**F1**	**Toxic**	**PPL**	**F1**
Baseline	None	0.545	3.08	0.193	0.496	1.94	0.225	0.488	4.61	0.231	0.507	1.76	0.231
	Steering with probe	0.310	3.19	0.191	0.335	2.72	0.187	0.260	5.52	0.228	0.350	2.23	0.220
	DPO	0.210	3.15	0.195	0.241	2.69	0.221	0.245	5.15	0.228	0.221	2.01	0.233
Activation patching	Patch all four groups	0.139	3.08	0.169	0.278	1.94	0.207	0.260	4.58	0.213	0.138	1.78	0.209
	Patch all ↑ neurons	0.853	6.05	0.154	0.536	2.64	0.184	0.686	4.58	0.199	0.611	1.78	0.199
Activation editing (probe-based, descending cossim)	$\alpha = 0.01, \beta = 0.8$	0.123	3.08	0.179	0.045	2.19	0.186	0.199	4.54	0.188	0.038	1.77	0.179
	$\alpha = 0.01, \beta = 0.6$	0.159	3.08	0.181	0.183	2.11	0.193	0.200	4.56	0.201	0.098	1.77	0.196
	$\alpha = \mathbf{0.01}, \beta = \mathbf{0.55}$	0.203	3.08	0.183	0.241	1.96	0.196	0.216	4.56	0.210	0.125	1.77	0.202
	$\alpha = 0.05, \beta = 0.5$	0.211	3.08	0.184	0.299	1.96	0.200	0.260	4.56	0.204	0.264	1.77	0.197
Activation editing (probe-based, ascending activation)	$\alpha = 0.01, \beta = 0.8$	0.025	3.08	0.158	0.097	2.39	0.188	0.271	4.56	0.183	0.154	1.77	0.196
	$\alpha = \mathbf{0.01}, \beta = \mathbf{0.6}$	0.075	3.07	0.178	0.204	2.26	0.198	0.295	4.57	0.202	0.218	1.77	0.201
	$\alpha = 0.01, \beta = 0.55$	0.111	3.08	0.175	0.258	2.25	0.203	0.330	4.57	0.199	0.229	1.77	0.202
	$\alpha = 0.05, \beta = 0.5$	0.109	3.08	0.178	0.310	1.96	0.204	0.331	4.58	0.204	0.251	1.77	0.193



GPT-2-Medium

Gemma-2-2B

Mistral-7B

Figure 6: *Four neuron groups collectively reduce toxicity during DPO, shown for GPT-2-Medium, Gemma-2-2B, and Mistral-7B.* The same four groups emerge consistently across models. (a) Proportion of toxicity reduction per group, where TP ↓ and AN ↓ dominate; (b) Cumulative toxicity reduction for the top 40,000 neurons (ranked by projection reduction), where TP ↓ dominates early ranks and AN ↓ gradually catches up the effect; (c) Per-group activation shifts during DPO for the top 2,000–2,500 neurons, where each group shifts according to its orientation relative to the toxic probe.

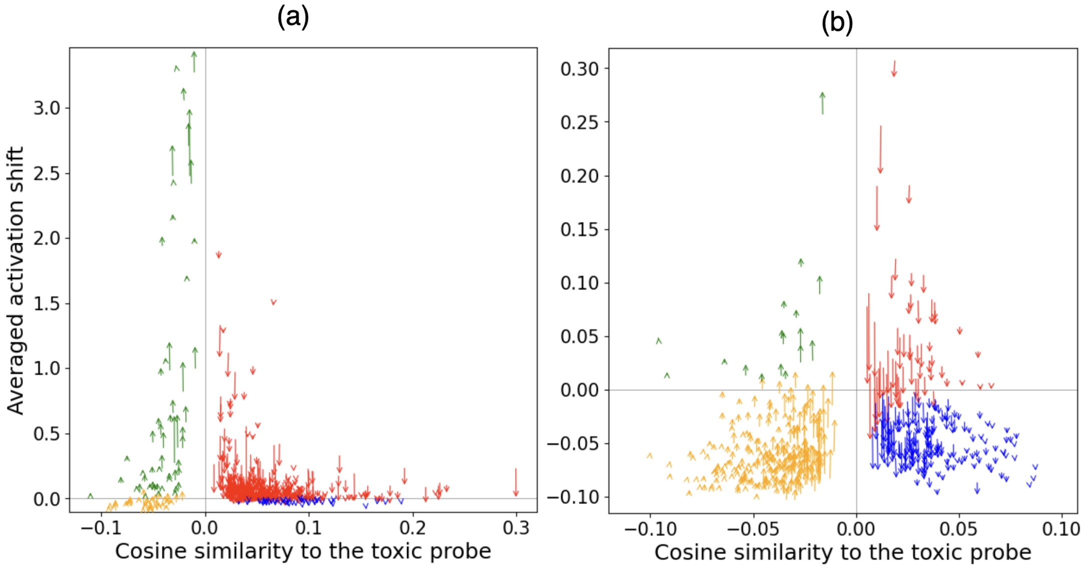


Figure 7: *Activation shifts of top-contributing neurons to toxicity projection reduction in GPT-2-Medium.* (a) Activation shifts of top 500 neurons, where TP ↓ drives the reduction. (b) Activation shifts of neurons ranked 5000–5500, showing increased AN ↓ influence and more balanced contributions across all four groups.

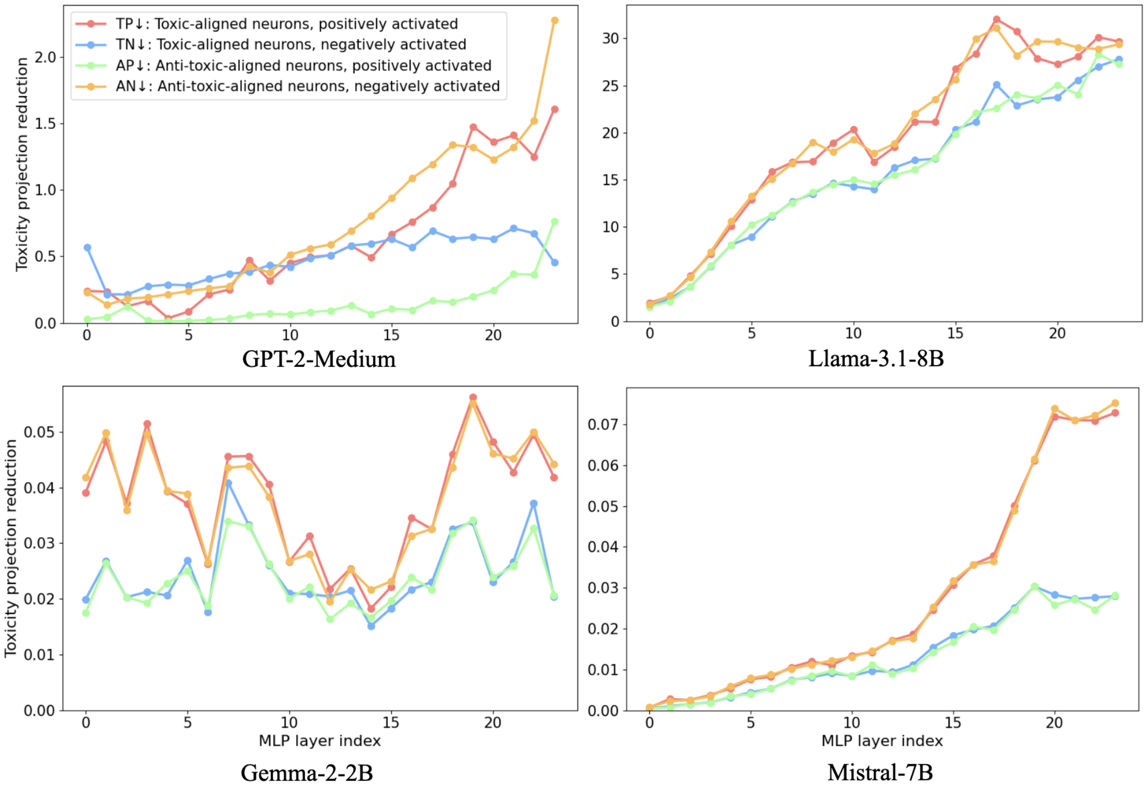


Figure 8: *Layer-wise toxicity projection reduction by neuron group.* Toxicity reduction generally increases across MLP layers under the cumulative group effects, though the upward trend is less evident for Gemma-2-2B. The upward trend shows that each layer progressively shifts away from toxicity, with the largest toxicity reduction occurring in later layers.