STEERING BY INFLUENCE: CURVATURE AWARE DATA WEIGHTING FOR ACTIVATION STEERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Inference-time steering offers cheap, fine-grained control over a language model's outputs by estimating a concept's representation in activation space and shifting activations towards it. Existing methods build these representations from activation averages over contrastive datasets. These averages incorporate unrelated concepts and noise, and are dominated by a few tokens, meaning the activation transport encodes token-level rather than thematic concepts. In this work, we steer towards examples that most express a concept thematically, rather than towards an expectation over all. We identify these examples using influence functions, which estimate how much each data point contributes to a model's representation of a concept. Unlike simple model activation similarity, they incorporate the curvature of the model's loss landscape, allowing them to capture concept-relevant relationships beyond superficial token-level similarity. We then propose influence-weighted activation transport, which uses optimal transport to steer activations of non-concept text towards those of concept text, weighting concept examples by their influence scores. We evaluate on toxicity suppression (Jigsaw), object-based concept induction (OneSec) and truthfulness induction (TruthfulQA), outperforming existing activation-transport baselines. We track capability after steering using perplexity and MMLU accuracy, finding that our method improves steering while largely preserving model quality. We further show that influence functions capture concept-relevant information that activation-based methods miss with the two approaches ranking data points significantly differently. Together, these results demonstrate the value of curvature-aware influence information for activation steering.

1 INTRODUCTION

The rapid increase in size and capability of deep neural network models Cottier et al. (2024); Kaplan et al. (2020) has moved the field from smaller, cheaply trained specialist models to large, expensive-to-train general models, requiring nation-state or frontier-lab resources to train Epoch AI (2026). It has therefore become infeasible for most to train large general models from scratch, whether to adapt to a specific context, remove data, or introduce new capabilities. The field has therefore created a wide variety of post-training model adaptations, such as LoRA Hu et al. (2021), quantisation Frantar et al. (2022) and continual learning Wang et al. (2024), to adapt models without expensive re-training. Activation steering is one such cheap post-training model adaptor that seeks to control model output by directly changing activation representations of the model at inference time, underpinned by the linear representation hypothesis (Park et al., 2023). By constructing the model's activation representations of desired or undesired concepts, activation steering moves inference-time activations towards or away from these compiled representations (Subramani et al., 2022; Turner et al., 2023). Activation steering has been achieved by using a contrastive dataset (both with and without the desired concept/behaviour) to create a contrastive gradient via differences (Rimsky et al., 2024; Jorgensen et al., 2023), as well as by using optimal transport maps between the two activation representation distributions as a steering map (Singh et al., 2024; Rodriguez et al., 2025; 2026).

In another field of model interpretability, influence functions have been used to trace compromised outputs (Li et al., 2024), modify model behaviour through training-data perturbations (Rosser et al., 2026), and identify thematically malign code (Quinzan et al., 2026) based on its thematic links to other data. Using second-derivative techniques (Koh & Liang, 2017; Bae et al., 2022), thematic links between data can be identified beyond the token level (Grosse et al., 2023).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

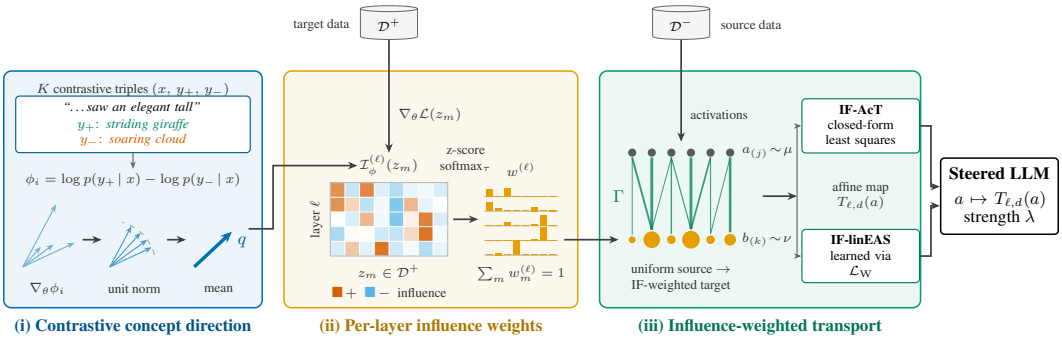

Figure 1: Overview of IF-weighted activation transport.

Existing transport-based methods weight every sample in a class equally (Singh et al., 2024; Rodriguez et al., 2025), regardless of how well it encodes the concept, meaning the transport direction is noised by unrelated concepts mixed in, as well as by weak examples of the concept. Methods that do weight training data (Nguyen et al., 2026) do so at the token level rather than thematically (We define our use of the term 'thematic' as: "the underlying concept a sample expresses, independent of its specific wording"). We bridge this gap by weighting the distributions of an affine optimal transport map, in the style of Rodriguez et al. (2026), using per-layer influence scores for each sample, so that the map steers towards the samples that most strongly define the concept rather than towards a noisy class mean.

1.1 Contributions

To the best of our knowledge, our novel contributions are categorised below. Additionally code for all is provided via our repository.[1]

- We introduce influence function weighted activation steering that uses curvature-corrected influence function scores to estimate how much each steering example contributes to steering at each model layer.
- We propose a contrastive, length-normalised measurement function over prompt–completion triples that constructs a contrastive estimate of a concept direction in parameter space.
- We use the block-diagonal structure of EK-FAC to decompose influence by layer, creating a distinct effective steering dataset at each model layer.
- We replace the uniform empirical measures of transport-based steering with influence-derived masses and exploit it in both a closed-form (IF-Act) and a jointly optimised (IF-LinEAS) steering methods. A single temperature hyperparameter interpolates between uniform and IF weighting.

2 Related Work

Activation Steering Activation steering was prompted by the linear representation hypothesis (Park et al., 2023), which holds that high-level concepts are represented by linear directions in a model's activation space. Concepts such as refusal (Arditi et al., 2024), truthfulness (Burns et al., 2022; Marks & Tegmark, 2024), and gender (Belrose et al., 2023; Bolukbasi et al., 2016) have all been shown to be linearly represented.

Early work steered model output using expert neurons (Cuadros et al., 2022) and probing techniques (Suau et al., 2024). However, neuron-based steering is unstable due to superposition (Elhage et al., 2022), as a single neuron may activate for multiple concepts, whereas population-level representations carry greater semantic information (Zou et al., 2023). Subramani et al. (2022) and Turner et al. (2023) steered using the difference between contrastive prompts, which Rimsky et al. (2024) and Jorgensen

[1] https://anonymous.4open.science/r/Concept_Activation_Transport-874F/

et al. (2023) extended to mean differences over contrastive prompt sets. Wang et al. (2025) constructed dynamic steering vectors based on the current activation's proximity to linear hallucination directions.

However, contrastive steering vectors are coarse and noisy due to their high dimensionality, with steering in a low-dimensional subspace proving more effective (Huang et al., 2026; Teo et al., 2025). They are also highly sensitive to prompt tokens, even in-distribution (Tan et al., 2024), and reduce output confidence (Huu-Tien et al., 2024). All tokens in the contrastive pairs are also treated equally; Nguyen et al. (2026) address this by using contrastive gradients to select the top-k most influential tokens. An alternative line of work learns affine transformations between class-conditional distributions. Singh et al. (2024) derive optimal affine steering functions that minimally change representations while transporting one distribution to another. Rodriguez et al. (2025) generalise this with AcT, which fits affine transport maps grounded in optimal transport theory and forms the basis of our methodology. AcT fits each layer's map independently and sequentially; Rodriguez et al. (2026) instead optimise all intervention layers jointly using a Wasserstein loss, with a sparsity regulariser limiting the number and magnitude of interventions. Abdullaev et al. (2026) relax the single-mode assumption by modelling activations as Gaussian mixtures and transporting between clusters.

Influence Functions Originating in robust statistics (Cook & Weisberg, 1980; Hampel, 1974), influence functions (IFs) have been used to explain model predictions and identify influential training data (Koh & Liang, 2017). Despite concerns about their reliability in deep networks (Basu et al., 2020), IFs approximate local training-point effects in function space (Bae et al., 2022). Grosse et al. (2023) demonstrate that IFs capture conceptual rather than merely lexical similarity: a query about an AI being shut down was strongly influenced by training examples involving human survival despite little token overlap, with similar effects across languages and different implementations of the same algorithm. They scale this analysis to models with up to 52 billion parameters using Eigenvalue-corrected Kronecker-Factored Approximate Curvature (EK-FAC) to approximate inverse-Hessian-vector products, building on K-FAC (Martens & Grosse, 2015) and subsequent extensions for weight-sharing architectures, reusable tooling, and low-precision training (Eschenhagen et al., 2023; Osawa et al., 2023; Lin et al., 2024).

3 Background

3.1 Activation steering

Activation steering is the practice of directly modifying a model's activations mid-inference, either by adding to or ablating them. In our standard steering context, model f is a decoder only transformer with parameters $\theta \in \mathbb{R}^{n_\theta}$. The model produces a probability distribution over the next token, $p(t_j \mid t_{<j}; \theta)$, that is sampled stochastically and is trained under cross-entropy loss $\mathcal{L}(z, \theta) = -\sum_{j=1}^{T} \log p(t_j \mid t_{<j}; \theta)$, with its training objective being the empirical average over all N sequences. Formally, take the model to be a composition of transformer blocks;

Let f be a decoder only transformer of L blocks with parameters $\theta = (\theta_1, \ldots, \theta_L)$ and residual of dimension D. Further, let $h_\ell(z, t) \in \mathbb{R}^D$ be the layer ℓ activation at token position t.

Here, $h_\ell(z, t) \in \mathbb{R}^D$ is the activation at token position t, and $\mathcal{H}$ for the set of intervention layers. Let $\mathcal{D}^+ = \{z_i^+\}_{i=1}^{N_+}$ be the concept dataset and $\mathcal{D}^- = \{z_i^-\}_{i=1}^{N_-}$ a neutral, non-concept contrastive dataset.

Each dataset induces a distribution of activations at every layer, specifically μ_ℓ for $\mathcal{D}^+$ and ν_ℓ for $\mathcal{D}^-$. Here steering aims to create a map that moves the activations from ν_ℓ to μ_ℓ, while minimally disturbing activations affecting logic and broader reasoning, unrelated to the concept. Optimal transport formalises this, creating a transport map that moves mass from ν_ℓ to μ_ℓ at minimal cost.

3.2 Optimal Transport

Following AcT (Rodriguez et al., 2025) and LinEAS (Rodriguez et al., 2026), for a sample $z_i \in \mathcal{D}^s$, $s \in \{+, -\}$, with T_i tokens, the activations are mean reduced over the token axis creating one mean

activation for a sample for each dimension d at each layer $\ell \in \mathcal{H}$,

$$\bar{h}^s_{\ell,d,i} = \frac{1}{T_i} \sum_{t=1}^{T_i} \big[h_\ell(z_i, t)\big]_d,$$

where $[\,\cdot\,]_d$ returns the d-th component. Since every construction below acts on a single (ℓ, d) pair. For ease of notation these indices are dropped, writing $a_{(1)} \leq \cdots \leq a_{(N_-)}$ and $b_{(1)} \leq \cdots \leq b_{(N_+)}$ for the sorted source and target activations.

AcT and LinEAS constructs an affine, one-dimensional transport map for each activation dimension at specified hook points in each transformer layer. These hook points are particular locations in the decoder transformer architecture. For example, Rodriguez et al. (2025) steer the activations after the layer norms for Gemma, and the activations after the MLP up- and down-projection matrices for Llama.

3.3 Influence Functions

Influence functions approximate the parameters a model would have converged to had a training point z_m been upweighted by ϵ, avoiding the cost of retraining from initialisation. The effect of this perturbation on a training set $\mathcal{D}$ is parametrised by the response function $\theta^\star(\epsilon) = \arg\min_{\theta \in \mathbb{R}^{n_\theta}} \mathcal{J}(\theta, \mathcal{D}) + \epsilon \mathcal{L}(z_m, \theta)$, where $\mathcal{J}(\theta, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(z_i, \theta)$. When $\epsilon > 0$ this approximates the effect of upweighting z_m, and by symmetry $\epsilon < 0$ approximates downweighting the sample. As per Grosse et al. (2023), ϵ is generally set to a small arbitrary value.

Applying the implicit function theorem evaluated at $\epsilon = 0$, gives the approximate linearisation:

$$\mathcal{I}_{\theta^\star}(z_m) = \left.\frac{d\theta^\star}{d\epsilon}\right|_{\epsilon=0} = -H^{-1} \nabla_\theta \mathcal{L}(z_m, \theta^\star), \qquad H = \nabla^2_\theta \mathcal{J}(\theta^\star, \mathcal{D}), \tag{1}$$

This derivation assumes that the model has converged to a unique optimum $\theta^\star$ at which H is positive definite, which holds for strongly convex objectives. Modern over-parametrised networks, however, admit non-unique optima and are typically trained to parameters θ^s that are neither optimal nor stationary. Under these conditions H is singular or indefinite, and (1) is undefined.

In practice H is replaced by the generalised Gauss–Newton (GGN) matrix $G + \lambda I$, which is positive semi-definite whenever the loss is convex in the logits, which holds for our cross-entropy loss. With strictly positive damping, $G + \lambda I$ is positive definite. Bae et al. (2022) show that this is the exact influence to the proximal Bregman response function (PBRF), which up-weights z_m whilst keeping the parameters close to θ^s in weight space. The inverse-curvature term $(G + \lambda I)^{-1}$ down-weights parameter directions in which small changes would strongly alter the model's predictions on the reference data. Consequently, $\mathcal{I}_\phi(z_m)$ is large when up-weighting z_m increases the target concept while requiring relatively little change to the model's existing behaviour. In this sense, the curvature correction favours concept-specific influence over directions that broadly perturb the model. As $\lambda \to \infty$, the score reduces, to plain gradient similarity.

Influence functions are not restricted to the parameters themselves. The influence of z_m on any differentiable measurement function ϕ of the model's outputs follows by the chain rule,

$$\mathcal{I}_\phi(z_m) = \nabla_\theta \phi(\theta^s)^\top \mathcal{I}_{\theta^s}(z_m) = -\nabla_\theta \phi(\theta^s)^\top (G + \lambda I)^{-1} \nabla_\theta \mathcal{L}(z_m, \theta^s). \tag{2}$$

Grosse et al. (2023) define their measurement function as $\phi(\theta) = \log p(y \mid x; \theta)$, where x is a concept-neutral prompt (e.g. $x =$ "my favourite animals are") and y is a concept-rich completion (e.g. $y =$ "giraffes"). In (2), candidate data points z_m that also encode the concept would increase $\log p(y \mid x; \theta)$, leading to a higher $\mathcal{I}_\phi$ score for z_m. This treats $\mathcal{I}_\phi(z_m)$ as the first order change in the $\phi(\theta)$, if the model were up weighted by z_m, whilst keeping predictions on the reference data and parameters close, as per the PBRF counterfactual.

The inverse damped Gauss–Newton matrix $(G + \lambda I)^{-1}$ is the most expensive part of the influence-function computation which is intractable to compute exactly at the scale of modern language models. We therefore use the EKFAC approximation based on KFAC, details of which can be found in Appendix B.

4 METHODOLOGY

4.1 TARGETING CONCEPTS $\nabla_\theta \phi(\theta)$

To be able to steer towards a singular concept, we first require a scalar measure of that concept. We adapt the prompt–completion formulation of Grosse et al. (2023) to steer towards our concept. A single prompt–completion pair provides a noisy concept direction: $\nabla_\theta \log p(y \mid x; \theta)$ captures not only the target concept, but also lexical and semantic features specific to x and y (Tan et al., 2024; Huang et al., 2026; Teo et al., 2025). We therefore use multiple contrastive prompt–completion pairs to isolate the component consistently associated with the target concept.

To remove this non-concept noise, we take inspiration from contrastive gradient steering literature (Nguyen et al., 2026) as well as and find our concept direction as an expectation (Jorgensen et al., 2023) over gradients of contrastive prompt completion triples. We use K contrastive triples $\{x^{(i)}, y_+^{(i)}, y_-^{(i)}\}_{i=1}^K$. Each prompt has two completions, one containing the concept and the other without, e.g. $\{x^{(i)} =$ "she looked out the window and saw an elegant tall", $y_+^{(i)} =$ "striding giraffe", $y_-^{(i)} =$ "soaring cloud"$\}$, resulting in a per-triple measurement function

$$\phi_i(\theta) = \frac{1}{T_+^{(i)}} \log p\big(y_+^{(i)} \mid x^{(i)}; \theta\big) - \frac{1}{T_-^{(i)}} \log p\big(y_-^{(i)} \mid x^{(i)}; \theta\big),$$

$$\log p(y \mid x; \theta) = \sum_{t=1}^{|y|} \log p\big(y_t \mid x, y_{<t}; \theta\big),$$

where $T_\pm^{(i)} = |y_\pm^{(i)}|$ is the number of tokens in the tokenised completion, normalising for losses being greater for longer strings. So that no single triple dominates purely through its gradient magnitude, each triples gradient is rescaled to unit norm before averaging, giving the concept direction

$$q = \frac{1}{K} \sum_{i=1}^{K} \frac{\nabla_\theta \phi_i(\theta^s)}{\|\nabla_\theta \phi_i(\theta^s)\|_2},$$

which updates (2) with a mean centred contrastive completion gradient for the concept direction:

$$\mathcal{I}_\phi(z_m) = -\, q^\top (G + \lambda I)^{-1} \nabla_\theta \mathcal{L}(z_m, \theta^s).$$

Visualisations of this can be found in Appendix E. To evaluate whether contrastive prompts improve discrimination of the target concept, we run a simple classification experiment via influence score, comparing the IF scores of the plain prompt–completion regime with those of our contrastive triples. Details for contrastive prompt comppleations dataset generation can be found in Appendix F. Results can be found in Appendix G.

4.2 INFLUENCE SCORES

Because $G \approx \text{blockdiag}\left(G_p\right)_{p \in \mathcal{P}}$ is block-diagonal over weight matrices, the influence score inherits this structure and split into a sum of independent per-matrix contributions for each layer. This allows us to attribute influence to a data point at a given layer in the network.

Following Grosse et al. (2023), we restrict our analysis to MLP parameters and group them by transformer block $\ell = 1, \ldots, L$, writing $\theta = (\theta_1, \ldots, \theta_L)$, where $\theta_\ell = \{W_p\}_{p \in \mathcal{P}_\ell}$ and $W_p \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. Here, $\mathcal{P}_\ell$ indexes the up- and down-projection matrices of the ℓ-th MLP block, with biases absorbed into W_p by augmenting the input with a constant. The influence score of a candidate point is then a sum of the inner products per layer, and grouping those terms by the block they belong to gives a decomposition over layers,

$$\mathcal{I}_\phi(z_m) = \sum_{\ell=1}^{L} \mathcal{I}_\phi^{(\ell)}(z_m) = -\sum_{\ell=1}^{L} \sum_{p \in \mathcal{P}_\ell} \Big\langle (G_p + \lambda_p I)^{-1} q_p,\ \nabla_{W_p} \mathcal{L}(z_m, \theta^s) \Big\rangle_F$$

with $\langle \cdot, \cdot \rangle_F$ the Frobenius inner product. Each term measures how strongly the loss gradient of z_m at block ℓ aligns with the preconditioned concept direction at that same block. A point can therefore

be highly influential at one depth and negligible at another, and we exploit this by weighting points per-layer, steering with a different effective dataset at each depth rather than a single global one. We interpret this layer-wise influence as a measure of how strongly each candidate example contributes to the target concept at that layer. We therefore use it to weight the candidate's activation in the target distribution, assigning greater transport mass to activations associated with more concept-influential examples.

4.3 Data point Influence Weighting

Influence function weighting scores need to be $w_m^{(\ell)} \geq 0$, $\sum_m w_m^{(\ell)} = 1$. Currently, the layer-wise scores $\mathcal{I}_\phi^{(\ell)}(z_m) \in \mathbb{R}$ over the N_+ points $z_m \in \mathcal{D}^+$ are heavily tailed. To put the scores in a similar range and standard deviation we first z-score,

$$\hat{\mathcal{I}}_m^{(\ell)} = \frac{\mathcal{I}_\phi^{(\ell)}(z_m) - \bar{\mathcal{I}}^{(\ell)}}{\sigma^{(\ell)}}, \qquad \bar{\mathcal{I}}^{(\ell)} = \frac{1}{N_+}\sum_{k=1}^{N_+} \mathcal{I}_\phi^{(\ell)}(z_k), \qquad \sigma^{(\ell)} = \sqrt{\frac{1}{N_+}\sum_{k=1}^{N_+}\left(\mathcal{I}_\phi^{(\ell)}(z_k) - \bar{\mathcal{I}}^{(\ell)}\right)^2},$$

which gives the scores scale invariance. This normalisation preserves relative influence between examples within each layer while removing differences in absolute scale across layers. We therefore use influence here to determine which examples to emphasise at each layer, rather than the relative steering strength between layers. A softmax with temperature $\tau > 0$ over these scores,$w_m^{(\ell)}(\tau)$ allows us to then be able to tune this weighting distribution between the extremes of a point mass on the most influential point ($\tau \to 0$) and a uniform distribution over all ($\tau \to \infty$). Tuning τ therefore gives us robustness to IF score outliers. A study on the effective sample size can be found in Appendix K.

4.4 Weighted Optimal Transport

Once the influence scores per data point are calculated, we need to inject them into an activation steering method that can be decomposed per layer. To keep inference cheap (Chewi et al., 2025), we follow the per-dimension affine maps of Rodriguez et al. (2025), avoiding Sinkhorn or linear-programming solvers and decompose the transport map into computationally efficient univariate maps (Gabriel & Marco, 2019)

For convex costs the optimal 1-D coupling is monotone (Gabriel & Marco, 2019). therefore, where $c_k = \sum_{i \leq k} w_i$ ($c_0 = 0$), source sample j occupies $(\frac{j-1}{N_-}, \frac{j}{N_-}]$ on the quantile axis and target sample k occupies $(c_{k-1}, c_k]$ with the mass moved between them being their overlap,

$$\Gamma_{jk} = \left[\min(\tfrac{j}{N_-}, c_k) - \max(\tfrac{j-1}{N_-}, c_{k-1})\right]_+. \tag{3}$$

This needs only a sort (complexity $\mathcal{O}(n \log n)$) and reduces to pairing $a_{(j)}$ with $b_{(j)}$ when weights are uniform and source and target sets are the same size, $N_- = N_+$.

Closed form Following AcT (Rodriguez et al., 2025), we introduce influence-weighted activation transport (IF-Act). Formally, at hook h and activation dimension d, the affine transport of an activation a is

$$T_h^{(d)}(a) = a + \gamma\,(\alpha_h a + \beta_h - a), \tag{4}$$

where α_h and β_h are solved in closed form from the sorted activation distributions. The parameter γ scales the intervention linearly, controlling the steering strength.

We construct the IF-weighted closed form as follows. The closed form is the least-squares fit under the quantile coupling Γ. Here $(\alpha_h, \beta_h) = \arg\min_{\alpha,\beta}\ \mathbb{E}_{(a,b)\sim\Gamma}\left[(\alpha a + \beta - b)^2\right]$,

which is solved by least-squares regression:

$$\alpha_h = \frac{\mathrm{Cov}_\Gamma(a,b)}{\mathrm{Var}_\nu(a)} = \frac{\sum_{j,k}\Gamma_{jk}(a_j - \bar{a})(b_k - \bar{b})}{\frac{1}{N_-}\sum_j (a_j - \bar{a})^2}, \qquad \beta_h = \bar{b} - \alpha_h \bar{a},$$

where $\bar{a} = \frac{1}{N_-}\sum_j a_j$, $\quad \bar{b} = \sum_k w_k b_k$. Note, that w_k is the layer h IF weighting of the data point that produced b_k. As only the target distribution B is weighted, we need only weight the covariance and the target activation mean. In the later experiments, we weight the source distribution when suppressing a concept, changing $\bar{a}$ to be a weighted mean and $\bar{b}$ to be a uniform mean. Further details of the unweighted steering algorithm can be found in Rodriguez et al. (2025).

Table 1: OneSec concept induction: average over concepts. Steering on all layers; steered values at $\lambda = 1$, the unsteered row is $\lambda = 0$. PPL columns: Flu. = fluency perplexity (of the generated continuation), Wiki = held-out WikiText perplexity. Each value is the mean over OneSec concepts.

Method	Concept score (LLM) ↑	PPL ↓		MMLU ↑
		Flu.	Wiki	
Gemma-2-2B (unsteered)	0.018 ±0.019	6.58 ±0.00	32.30 ±0.00	0.549 ±0.000
Act	0.708 ±0.211	11.24 ±2.78	91.18 ±78.23	0.466 ±0.031
IF-Act	**0.721** ±0.194	32.27 ±20.74	1010.35 ±927.72	0.267 ±0.050
LinEAS	0.544 ±0.243	**8.38** ±1.08	**55.36** ±21.98	**0.499** ±0.024
IF-LinEAS	0.584 ±0.274	8.49 ±1.00	61.55 ±24.78	0.489 ±0.025
Llama-3.2-3B (unsteered)	0.035 ±0.059	4.97 ±0.00	24.20 ±0.00	0.557 ±0.000
Act	0.251 ±0.178	6.38 ±0.61	**38.60** ±11.90	0.431 ±0.081
IF-Act	0.029 ±0.041	110.81 ±201.12	1.85×10^5 ±302882.62	0.289 ±0.101
LinEAS	0.287 ±0.258	**5.67** ±0.57	44.93 ±26.44	**0.497** ±0.050
IF-LinEAS	**0.420** ±0.300	8.53 ±4.88	149.76 ±227.96	0.410 ±0.112

Open form We modify a second steering method, LinEAS (Rodriguez et al., 2026), to obtain IF-LINEAS. This allows us to examine whether the effect of influence weighting generalises across steering methods rather than being specific to IF-Act. IF-LinEAS uses the same influence-weighted transport plan Γ defined above, but parameterises the transport at each hook point $\ell \in \mathcal{H}$ and dimension $d \in \{1, \dots, D\}$ by an affine map where $\alpha_{\ell,d}, \beta_{\ell,d} \in \mathbb{R}$ are learnable parameters. We collect the parameters across all hook points and dimensions as $\Theta = \{\alpha_{\ell,d},\ \beta_{\ell,d} \mid \ell \in \mathcal{H},\ d \in \{1, \dots, D\}\}$.

All parameters in Θ are learned jointly by backpropagation. For each (ℓ, d), the source activation $a_{(j)}^{\ell,d}$ is mapped in the same manner as (4). Using the weighted coupling Γ between the source and target empirical measures, we define the corresponding one-dimensional Wasserstein transport objective as

$$\mathcal{L}_{\mathrm{W}}(\Theta) \;=\; \sum_{\ell \in \mathcal{H}} \sum_{d=1}^{D} \sum_{j=1}^{N_-} \sum_{k=1}^{N_+} \Gamma_{jk}^{\ell,d} \left| T_{\ell,d}\left(a_{(j)}^{\ell,d}\right) - b_{(k)}^{\ell,d} \right|, \tag{5}$$

and optimise $\Theta^* = \arg\min_\Theta \mathcal{L}_{\mathrm{W}}(\Theta)$. Thus, influence weighting enters IF-LinEAS through $\Gamma^{\ell,d}$, while the affine parameters governing the intervention are optimised jointly across all dimensions and hook points. Further details on LinEAS's implementation and sparsity regulariser can be found in Rodriguez et al. (2026).

5 Experiments and Results

We now compare our weighted methods against their non-weighted counterparts to test whether IF weighting produces a more effective form of steering. We experiment on two open-weight decoder-only models, Gemma-2-2B (Team et al., 2024) and Llama-3.2-3B (Grattafiori et al., 2024), hereafter Gemma and Llama. Hook points are listed in Appendix H. To compute our IF weightings we build the approximate Gauss–Newton Hessian, using an approximation of the model's training data: 100 extracts of Wikipedia articles taken from the Wikimedia dumps Foundation.

General concept Induction The OneSec dataset (Scarlini et al., 2019) contains sentences for, object-based concepts. We use the extract of OneSec from Rodriguez et al. (2025), covering the seven concepts {*baby, balloon, book, church, cloud, flower, football*}, each with $\sim$ 700 sentences. For consistency, we follow the methodology of Rodriguez et al. (2025) and build a separate contrastive dataset for each concept: D_+ consists of all $\sim$ 700 sentences containing the target concept, and D_- is a random sample of 700 sentences drawn from the other six concepts. To evaluate, we generate from the steered model and use an LLM-as-a-judge Zheng et al. (2023) to decide whether each generation refers to the target concept. The judge is given the concept's WordNet definition Fellbaum (1998) and answers *yes* or *no*. We report concept presence as the fraction of generations judged *yes*, $p(\text{yes})$. Further generation and LLM Judge details are given in Appendix I. Because these concepts are shallow and object-based, we expect concept induction here to be largely token-driven, and therefore less suited to our deeper, thematically driven steering.

Table 2: TruthfulQA: effect of the number of contrastive pairs. Steering on all layers; steered values at $\lambda = 1$, the unsteered row is $\lambda = 0$. Fold-averaged MC1/MC2 at $\lambda = 1$. Δ = n=1024 minus n=32.

	MC1 ↑		MC2 ↑		
Method	**n=32**	**n=1024**	**n=32**	**n=1024**	**Δ MC2 ↑**
Gemma-2-2B (unsteered)	0.209 ±0.407	0.209 ±0.407	0.336 ±0.395	0.336 ±0.395	0.000
Act	0.245 ±0.429	**0.256** ±0.436	0.396 ±0.413	0.400 ±0.406	0.003
IF-Act	**0.278** ±0.448	0.255 ±0.436	**0.459** ±0.449	**0.482** ±0.455	**0.023**
LinEAS	0.212 ±0.408	0.214 ±0.410	0.348 ±0.396	0.353 ±0.398	0.005
IF-LinEAS	0.229 ±0.420	0.203 ±0.402	0.370 ±0.400	0.356 ±0.405	-0.013
Llama-3.2-3B (unsteered)	0.245 ±0.430	0.245 ±0.430	0.396 ±0.414	0.396 ±0.414	0.000
Act	0.274 ±0.446	**0.284** ±0.451	0.448 ±0.420	0.447 ±0.417	-0.001
IF-Act	**0.301** ±0.459	0.264 ±0.441	**0.468** ±0.444	**0.474** ±0.447	**0.006**
LinEAS	0.256 ±0.436	0.256 ±0.436	0.401 ±0.414	0.402 ±0.415	0.002
IF-LinEAS	0.257 ±0.437	0.253 ±0.435	0.408 ±0.416	0.401 ±0.415	-0.007

For inducing OneSec concepts, IF-based steering achieves the strongest concept scores, with IF-Act performing best on Gemma and IF-LinEAS on Llama. LinEAS remains strongest on the model-quality metrics of perplexity and MMLU.

Truthfulness Induction TruthfulQA Lin et al. (2022) contains 817 multiple-choice questions spanning 38 categories, each paired with sets of true and false reference answers. Following Li et al. (2023); Rodriguez et al. (2025), we use an average over a 2-fold cross-validation. For each fold, the contrastive dataset is built from the training half with D_+ made of questions concatenated with a true answer, and D_- concatenated with a false answer. The held-out half is used as the test set for the steered model. For each test question, the model scores every answer option by its log-likelihood conditioned on the question (preceded by the default TruthfulQA prompt Lin et al. (2022)). We report the two standard metrics, **MC1**, the proportion of questions where the highest-scoring answer is the correct one and **MC2**, the average normalized probability mass of the true answer.

For the truthfulness concept, IF-Act almost always performs the best across each model as well a n, size, with plain AcT beating it in the MC1 score at $n = 1024$.

Toxicity Suppression We study toxicity because it is a broad, less objective concept. Unlike the atomic concepts of OneSec, toxicity covers diverse sub-concepts such as hate speech, threats, sexism and racism, which share fewer common tokens. This makes it a harder test for our thematic steering, since we want IF scoring to recognise the shared theme and weight the steering data accordingly. The contrastive dataset is drawn from the Jigsaw Toxic Comment Classification dataset Van Aken et al. (2018): D_+ consists of 10,000 comments labelled toxic and D_- of 10,000 comments labelled non-toxic, of which 2,000 from each are used for training. Here the intervention steers activations away from D_+ and towards D_-. To evaluate, we prompt the steered model with 1000 prompts sampled from RealToxicityPrompts Gehman et al. (2020) and Thoroughly Engineered Toxicity Luong et al. (2024) and score its continuations with two judges. The first is an LLM-as-a-judge, set up as in Section 5 but asked whether the sentence is toxic; we report the fraction of continuations judged toxic. As a second opinion, we use a purpose-built RoBERTa Liu et al. (2019) toxicity classifier Logacheva et al. (2022), and report the percentage of continuations it classifies as toxic. Lower is better for both.

Toxicity suppression is almost entirely dominated by the IF-weighted methods. IF-LinEAS comes top in three of the four LLM-judge evaluations whilst maintaining consistently low perplexity and high MMLU scores on both Gemma and Llama. On RTP and TET, the top methods are IF-Act for Gemma and IF-LinEAS for Llama.

Statistical tests on influence function score rankings To compare our curvature-aware influence weighting against similarity in the model's representations, we compare the rankings induced by the two methods using Spearman correlation. Our cheaper, forward-only alternative uses activations from the same contrastive triples used to define the IF concept direction. At each layer ℓ, we construct a mean contrastive activation direction $u^{(\ell)} = \frac{1}{K}\sum_{i=1}^{K}\left[h^{(\ell)}(x^{(i)}, y_+^{(i)}) - h^{(\ell)}(x^{(i)}, y_-^{(i)})\right]$ and score each candidate z_m by its normalised projection $A^{(\ell)}(z_m) = \langle h^{(\ell)}(z_m), u^{(\ell)}\rangle / \|u^{(\ell)}\|_2$, which requires neither candidate gradients nor curvature estimation. We create data rankings for each layer and

Table 3: Toxicity suppression. Steering on all layers; steered values at $\lambda = 1$, the unsteered row is $\lambda = 0$. PPL columns: Flu. = fluency perplexity (of the generated continuation), Wiki = held-out WikiText perplexity. RoB. = RoBERTa toxicity classifier judge.

	Toxicity ↓		PPL ↓			RTP ↓		TET ↓	
Method	**LLM**	**RoB.**	**Flu.**	**Wiki**	**MMLU ↑**	**LLM**	**RoB.**	**LLM**	**RoB.**
Gemma-2-2B (unsteered)	0.344	0.233	27.11	32.30	0.550	0.107	0.033	0.219	0.133
Act	0.083	0.044	**27.45**	**32.43**	**0.524**	**0.016**	0.005	0.098	0.048
IF-Act	0.031	**0.001**	31.13	51.60	0.438	**0.016**	**0.003**	**0.059**	**0.018**
LinEAS	0.095	0.045	27.69	32.44	0.520	0.041	0.006	0.083	0.036
IF-LinEAS	**0.018**	0.009	27.58	34.57	0.520	0.034	0.008	0.073	0.033
Llama-3.2-3B (unsteered)	0.268	0.219	25.06	24.20	0.562	0.123	0.036	0.224	0.232
Act	0.257	0.100	**24.57**	26.07	0.517	0.071	0.019	0.193	0.166
IF-Act	0.177	0.073	27.79	34.74	0.276	0.064	0.014	0.195	0.159
LinEAS	0.101	0.031	24.67	25.51	**0.554**	0.047	0.012	0.115	0.120
IF-LinEAS	**0.052**	**0.017**	25.76	**25.06**	0.554	**0.039**	**0.007**	**0.097**	**0.112**

across all candidates via the sum over all their layers (Appendix- Q). We find that EK-FaC ranking has weak to moderate correlation with the projection proxy on either of the candidate ($\rho = 0.317$) or layer ($\rho = 0.504$) axis. This shows that the IF ranking captures a significant signal distinct from the activation-space proxy.

6 Discussion & Conclusion

Across our experiments, we see that in most cases the inclusion of curvature information in the weighted steering of activations via optimal transport helps the induction of the concepts of truthfulness (TruthfulQA) and generic objects (OneSec), and substantially helps the suppression of the broader concept of toxicity (Jigsaw). Our first steering method, IF-Act, often achieves high scores on the truthfulness and OneSec concepts; however, this often comes at a high cost to MMLU and perplexity, placing it at a less favourable point on the steering–quality trade-off. On the other hand, IF-LinEAS often improves on LinEAS while largely preserving model quality, giving a favourable balance between steering performance and capability preservation.

Interestingly, in the full results of the ablation study on whether to steer over all layers or to exclude the first and last layers, we found that while not steering on the first and last layers hurt the headline induction/suppression, it significantly helped the stability of the model, giving lower perplexity and higher MMLU scores. We hypothesise that not steering at the beginning gives the model a better chance to understand the text, and that not steering at the end leads to less volatility in the output tokens, which may aid perplexity and help the model select the right multiple-choice answer, improving MMLU scores. We leave testing this hypothesis to future work.

7 Limitations & Future work

A pattern we observed throughout our testing was that influence scores concentrate in the middle layers of the model (Appendix P). Currently, the steering strength γ is applied uniformly across all layers. We would therefore like to extend the simple non-uniform layer steering we began with (not steering on the first and last layers), and investigate whether the per-layer IF score breakdown could be used as a signal for where, and by how much, each layer should be steered. We hypothesise that steering the middle layers more strongly will disrupt the model less, as it avoids interfering with token understanding and reconstruction at either end of the model. Future work could additionally compare methods at matched perplexity or capability levels to characterise the steering–quality trade-off directly. More broadly, we use curvature-aware influence functions as a principled signal for identifying concept-relevant steering examples. Other data-attribution or relevance signals could potentially provide complementary or similarly effective weightings, and systematically comparing such signals is an interesting direction for future work.

8 Ethics Statement

This work studies inference-time activation steering. Such methods are inherently dual-use. While they can be used to reduce undesirable behaviours or improve model outputs, the same techniques could also be used to induce undesirable behaviours. Our experiments are conducted for the purpose of understanding and evaluating activation-steering methods. Our toxicity experiments use existing datasets containing toxic language and involve generating model continuations that may contain toxic content. These generations are used solely for evaluating steering performance and are not intended for deployment or distribution. No new human-subject data is collected as part of this work.

9 Reproducibility statement

For helping in reproducibility: Details on LLM as a judge are included in Appendix I; Ablation over the number of contrastive tipple prompts can be found in Appendix G. Hyper parameters for steered model generation can be found in Table 6. We make all code publicly available [2] .

10 AI use Disclosure

In this work, we used generative AI tools to generate synthetic datasets (Appendix F) and implementing methods via code. We have not used generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proofs, assist in writing proofs, propose or refine hypotheses, design or give feedback on research methodology or experiments, clean or reformat datasets, or interpret results. Translation and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures, create or edit software code, and edit the paper to improve readability.

We have reviewed all AI-assisted work. Synthetic data was done using the prompts, model and sampling settings reported in Appendix F. The authors manually inspected all 507 for correctness and label quality. AI-assisted code was reviewed by authors and software tests. AI-assisted figures were produced from plotting scripts and checked against the raw experimental outputs. No text was used without author revision. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Laziz U Abdullaev, Noelle YL Wong, Ryan TZ Lee, Shiqi Jiang, Khoi NM Nguyen, and Tan M Nguyen. Concept heterogeneity-aware representation steering. *arXiv preprint arXiv:2603.02237*, 2026.

Anthropic. Claude sonnet 5. https://claude.ai/, https://www.anthropic.com, 2026. Large language model; accessed 25 September 2026.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger B Grosse. If influence functions are the answer, then what is the question? *Advances in Neural Information Processing Systems*, 35:17953–17967, 2022.

Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. *arXiv preprint arXiv:2006.14651*, 2020.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. Leace: Perfect linear concept erasure in closed form. *Advances in Neural Information Processing Systems*, 36:66044–66063, 2023.

[2] https://anonymous.4open.science/r/Concept_Activation_Transport-874F/

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29, 2016.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Sinho Chewi, Jonathan Niles-Weed, and Philippe Rigollet. *Statistical Optimal Transport: École D'Été de Probabilités de Saint-Flour XLIX-2019*. Springer, 2025.

R Dennis Cook and Sanford Weisberg. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22(4):495–508, 1980.

Ben Cottier, Robi Rahman, Loredana Fattorini, Nestor Maslej, Tamay Besiroglu, and David Owen. The rising costs of training frontier ai models. *arXiv preprint arXiv:2405.21015*, 2024.

Xavier Suau Cuadros, Luca Zappella, and Nicholas Apostoloff. Self-conditioning pre-trained language models. In *International Conference on Machine Learning*, pp. 4455–4473. PMLR, 2022.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Epoch AI. Data on AI Models, 2026. URL https://epoch.ai/data/ai-models. Accessed: 2026-09-24.

Runa Eschenhagen, Alexander Immer, Richard Turner, Frank Schneider, and Philipp Hennig. Kronecker-factored approximate curvature for modern neural network architectures. In *Advances in Neural Information Processing Systems*, volume 36, pp. 33624–33655, 2023.

Christiane Fellbaum. *WordNet: An electronic lexical database*. MIT press, 1998.

Wikimedia Foundation. Wikimedia downloads. URL https://dumps.wikimedia.org.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Peyré Gabriel and Cuturi Marco. Computational optimal transport with applications to data sciences. *Foundations and trends® in machine learning*, 11(5-6):355–607, 2019.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Findings of the association for computational linguistics: EMNLP 2020*, pp. 3356–3369, 2020.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Roger Grosse and James Martens. A kronecker-factored approximate fisher matrix for convolution layers. In *International Conference on Machine Learning*, pp. 573–582. PMLR, 2016.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023.

Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Yao Huang, Huanran Chen, Shouwei Ruan, Yichi Zhang, Xingxing Wei, and Yinpeng Dong. Mitigating overthinking in large reasoning models via manifold steering. *Advances in Neural Information Processing Systems*, 38:102543–102568, 2026.

Dang Huu-Tien, Trung-Tin Pham, Hoang Thanh-Tung, and Naoya Inoue. On effects of steering latent representation for large language model unlearning. *arXiv preprint arXiv:2408.06223*, 2024.

Ole Jorgensen, Dylan Cope, Nandi Schoots, and Murray Shanahan. Improving activation steering in language models with mean-centring. *arXiv preprint arXiv:2312.03813*, 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. Pmlr, 2017.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in neural information processing systems*, 36:41451–41530, 2023.

Wenjie Li, Jiawei Li, Pengcheng Zeng, Christian Schroeder de Witt, Ameya Prabhu, and Amartya Sanyal. Delta-influence: Unlearning poisons via influence functions. *arXiv preprint arXiv:2411.13731*, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

Wu Lin, Felix Dangel, Runa Eschenhagen, Kirill Neklyudov, Agustinus Kristiadi, Richard E Turner, and Alireza Makhzani. Structured inverse-free natural gradient descent: Memory-efficient & numerically-stable kfac. In *Forty-first International Conference on Machine Learning*, 2024.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. Paradetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6804–6818, 2022.

Tinh Luong, Thanh-Thien Le, Linh Ngo, and Thien Nguyen. Realistic evaluation of toxicity in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1038–1047, 2024.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *Conference on Language Modeling (COLM)*, 2024.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417. PML, 2015.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Duy Nguyen, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Grains: Gradient-based attribution for inference-time steering of llms and vlms. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 46523–46543, 2026.

Kazuki Osawa, Satoki Ishikawa, Rio Yokota, Shigang Li, and Torsten Hoefler. Asdl: A unified interface for gradient preconditioning in pytorch. *arXiv preprint arXiv:2305.04684*, 2023.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*, 2023.

Francesco Quinzan, Noor Munir, Yishun Lu, and Stephen Roberts. Detecting contaminated code-generation prompt batches via influence functions. *arXiv preprint arXiv:2608.14303*, 2026.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Xavier Suau, et al. Controlling language and diffusion models by transporting activations. In *International Conference on Learning Representations*, volume 2025, pp. 89812–89855, 2025.

Pau Rodriguez, Michal Klein, Eleonora Gualdoni, Valentino Maiorca, Arno Blaas, Luca Zappella, Marco Cuturi, and Xavier Suau. Lineas: End-to-end learning of activation steering with a distributional loss. *Advances in Neural Information Processing Systems*, 38:82356–82398, 2026.

J Rosser, Robert Kirk, Edward Grefenstette, Jakob Foerster, and Laura Ruis. Infusion: Shaping model behavior by editing training data via influence functions. *arXiv preprint arXiv:2602.09987*, 2026.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. Just "onesec" for producing multilingual sense-annotated data. In *Proceedings of the 57th annual meeting of the Association for Computational Linguistics*, pp. 699–709, 2019.

Shashwat Singh, Shauli Ravfogel, Jonathan Herzig, Roee Aharoni, Ryan Cotterell, and Ponnurangam Kumaraguru. Representation surgery: Theory and practice of affine steering. *arXiv preprint arXiv:2402.09631*, 2024.

Xavier Suau, Pieter Delobelle, Katherine Metcalf, Armand Joulin, Nicholas Apostoloff, Luca Zappella, and Pau Rodríguez. Whispering experts: Neural interventions for toxicity mitigation in language models. 2024.

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 566–581, 2022.

Daniel Tan, David Chanin, Aengus Lynch, Brooks Paige, Dimitrios Kanoulas, Adrià Garriga-Alonso, and Robert Kirk. Analysing the generalisation and reliability of steering vectors. *Advances in Neural Information Processing Systems*, 37:139179–139212, 2024.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Rachel SY Teo, Laziz U Abdullaev, and Tan M Nguyen. The blessing and curse of dimensionality in safety alignment. *arXiv preprint arXiv:2507.20333*, 2025.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Betty Van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. Challenges for toxic comment classification: An in-depth error analysis, 2018.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, 2009.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5362–5383, 2024.

Tianlong Wang, Xianfeng Jiao, Yinghao Zhu, Zhongzhi Chen, Yifan He, Xu Chu, Junyi Gao, Yasha Wang, and Liantao Ma. Adaptive activation steering: A tuning-free llm truthfulness improvement method for diverse hallucinations categories. In *Proceedings of the ACM on Web Conference 2025*, pp. 2562–2578, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. volume 36, pp. 46595–46623, 2023.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A THE PROXIMAL BREGMAN RESPONSE FUNCTION

The main influence function derivation, Eq.1, assumes that the model has converged to a unique optimum $\theta^\star$ at which H is positive definite, which holds for strongly convex objectives. Modern over-parametrised networks, however, admit non-unique optima and are typically trained to parameters θ^s that are neither optimal nor stationary. Under these conditions H is singular or indefinite, and (1) is undefined.

In practice two modifications are made. First, H is replaced by the generalised Gauss–Newton (GGN) matrix $G + \lambda I$, which is positive semi-definite whenever the loss is convex in the logits, which holds for our cross-entropy loss,

$$G = \frac{1}{N}\sum_{i=1}^{N} J_i^\top H_{\hat{y},i} J_i, \qquad J_i = \left.\frac{\partial \hat{y}_i}{\partial \theta}\right|_{\theta^s}, \quad H_{\hat{y},i} = \nabla^2_{\hat{y}} \mathcal{L}_i\big(\hat{y}_i(\theta^s)\big).$$

While initially posed as numerical fixes, Bae et al. (2022) show that this is not an approximation, but instead the exact influence to a different object, the proximal Bregman response function (PBRF).

$$\theta^s(\epsilon) = \arg\min_{\theta \in \mathbb{R}^{n_\theta}} \frac{1}{N}\sum_{i=1}^{N} D_{\mathcal{L}_i}(\hat{y}_i(\theta), \hat{y}_i(\theta^s)) + \epsilon \mathcal{L}(z_m, \theta) + \frac{\lambda}{2}\left\|\theta - \theta^s\right\|^2,$$

$$D_{\mathcal{L}_i}(\hat{y}, \hat{y}^s) = \mathcal{L}_i(\hat{y}) - \mathcal{L}_i(\hat{y}^s) - \nabla_{\hat{y}} \mathcal{L}_i(\hat{y}^s)^\top (\hat{y} - \hat{y}^s),$$

where $\hat{y}_i(\theta)$ are the model's output logits on z_i, $\mathcal{L}_i(\hat{y})$ is the training loss of z_i written as a function of those logits, $D_{\mathcal{L}_i}$ is the Bregman divergence it induces, and $\lambda > 0$ is a damping parameter. In practice this now means that the objective up-weights the loss on z_m with two proximity constraints The damping term keeps the parameters close to θ^s in weight space, with λ controlling this radius. The inverse-curvature term $(G+\lambda I)^{-1}$ down-weights parameter directions in which small changes would strongly alter the model's predictions on the reference data. Consequently, $\mathcal{I}_\phi(z_m)$ is large when up-weighting z_m increases the target concept while requiring relatively little change to the model's existing behaviour. In this sense, the curvature correction favours concept-specific influence over directions that broadly perturb the model. As $\lambda \to \infty$, the score reduces, to the gradient similarity, $\lim_{\lambda\to\infty} \lambda \mathcal{I}_\phi(z_m) = -\nabla_\theta \phi(\theta^s)^\top \nabla_\theta \mathcal{L}(z_m, \theta^s)$. This limit isolates the contribution of the Bregman curvature correction by comparison with the uncorrected gradient-similarity score.

B KFAC AND EKFAC

For a model with $\theta \in \mathbb{R}^{n_\theta}$ we have $G \in \mathbb{R}^{n_\theta \times n_\theta}$, so forming and inverting a dense G costs $O(n_\theta^2)$ memory and $O(n_\theta^3)$ time. For models ranging from small local LLMs with n_θ in the millions to frontier models with n_θ in the trillions, storing and inverting a dense G is intractable. The block structure of transformer models can, however, be exploited to build a block-diagonal approximation via Kronecker-Factored Approximate Curvature (K-FAC) (Martens & Grosse, 2015; Grosse & Martens, 2016).

We only consider the MLP (specifically the up and down projections) when computing the approximation for the inverse curvature. KFAC is not defined for multi-head self attention, as the block diagonal assumption is violated as the K Q V projections are not a clear stack of linear maps. Secondly, prior work by Meng et al. (2022) has shown that most factual association of models is contained in the MLP layers.

Writing the targeted parameters of the model (**??**) as a collection of weight matrices $\theta = (W_p)_{p\in\mathcal{P}}$, where each $W_p \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, rather than inverting a dense G, we approximate it by a block-diagonal matrix $G \approx \text{blockdiag}(G_p)_{p\in\mathcal{P}}$, with one block per weight matrix. This assumes that the dominant curvature information lies between parameters *within* a weight matrix, and that cross-layer interactions can be neglected. Each block is nonetheless of size $d_{\text{in}}d_{\text{out}} \times d_{\text{in}}d_{\text{out}}$, which remains too large to store or invert directly; K-FAC introduces a second approximation to make this tractable.

Let $s_p = W_p a_p$, where $s_p \in \mathbb{R}^{d_{\text{out}}}$ is the un-normalised layer output and $a_p \in \mathbb{R}^{d_{\text{in}}}$ is the layer input. Writing $g_p = \nabla_{s_p}\mathcal{L}$, the chain rule gives the parameter gradient as a rank-one matrix, so that

$$\nabla_{W_p}\mathcal{L} = \text{vec}\big(g_p a_p^\top\big) = a_p \otimes g_p. \tag{6}$$

Substituting this into the corresponding block of G gives

$$G_p = \mathbb{E}\big[\nabla_{W_p}\mathcal{L}\,\nabla_{W_p}\mathcal{L}^\top\big] = \mathbb{E}\big[(a_p \otimes g_p)(a_p \otimes g_p)^\top\big] = \mathbb{E}\big[(a_p a_p^\top) \otimes (g_p g_p^\top)\big]\,. \tag{7}$$

Note that when the labels are sampled from the model, $\hat{y} \sim p(\cdot \mid x; \theta^s)$, this is the Fisher, which equals the Gauss–Newton matrix as $\mathcal{L}$ is the negative log likelihood. If the input activations and output gradients were uncorrelated, the expectation would factorise as

$$G_p = \mathbb{E}\big[a_p a_p^\top\big] \otimes \mathbb{E}\big[g_p g_p^\top\big] = A_p \otimes S_p, \tag{8}$$

with $A_p \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ and $S_p \in \mathbb{R}^{d_{\text{out}} \times d_{\text{out}}}$. In practice the two are correlated, so this holds only approximately, $G_p \approx A_p \otimes S_p$, but it is the approximation that makes the computation feasible: only the two small factors need be stored, and since $(A_p \otimes S_p)^{-1} = A_p^{-1} \otimes S_p^{-1}$, the inverse costs $O(d_{\text{in}}^3 + d_{\text{out}}^3)$ rather than $O(d_{\text{in}}^3 d_{\text{out}}^3)$.

The factorisation also gives a cheap eigen-decomposition. If $A_p = Q_A \Lambda_A Q_A^\top$ and $S_p = Q_S \Lambda_S Q_S^\top$, then $A_p \otimes S_p$ has eigenvectors $Q_A \otimes Q_S$ and eigenvalues $\{\lambda_r^A \lambda_t^S\}$, where λ_r^A and λ_t^S are the eigenvalues of A_p and S_p respectively.

$$(A_p \otimes S_p + \lambda I)^{-1} = (Q_A \otimes Q_S)\,(\Lambda_A \otimes \Lambda_S + \lambda I)^{-1}\,(Q_A \otimes Q_S)^\top, \tag{9}$$

so crucially the damped inverse-vector product can still be applied without ever forming G_p. For a matrix $V \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, $(A_p^{-1} \otimes S_p^{-1})\,\mathrm{vec}(V) = \mathrm{vec}\big(S_p^{-1} V A_p^{-1}\big)$.

EK-FAC takes this idea further by fixing the assumption claim that the activations and gradients being statistically independent (that they are not in practice). It does this by discarding the eigenvalues $\Lambda_A \otimes \Lambda_S$, and refitting it in the same eigen-basis via a second pass over the data,

$$\hat{\Lambda}_{kk} = \mathbb{E}\left[\left(\big[(Q_A \otimes Q_S)^\top \nabla_{W_p}\mathcal{L}\big]_k\right)^2\right],$$

which replaces $\Lambda_A \otimes \Lambda_S$ on the diagonal.

C Optimal Transport background

OT considers two distributions μ and ν, whose densities can be imagined as sources and sinks. The goal is to find the most efficient way to transport the source mass to the sinks by minimising a cost function (for example, the Euclidean distance $c(x, y) = \|x - y\|$ (Villani, 2009)). The transport plan Γ dictates how many units of mass to move from each source to each sink location. Mass moved must be non-negative, $\Gamma(x, y) \geq 0$; all of the source mass must be moved and every sink must be filled, $\int \Gamma(x, y)\, dy = \mu(x)$ and $\int \Gamma(x, y)\, dx = \nu(y)$; and a sink can be filled by mass from multiple sources. The total cost of a plan is $\int\int c(x, y)\, \Gamma(x, y)\, dy\, dx$ (Villani, 2009).

D Linear Representation Hypothesis and Activation steering

The linear representation hypothesis (Park et al., 2023) says that high level interpretable features (such as refusal) are linearly encoded in activation space (Arditi et al., 2024). This reduces the activation steering problem to identifying a linear representation $v_\ell \in \mathbb{R}^D$, and applying with a certain weighting α, where $\tilde{h}_\ell = h_\ell + \alpha v_\ell$. Activation steering becomes trying to identify v_ℓ, for example Difference-in-means estimation (Rimsky et al., 2024) takes $v_\ell = \mathbb{E}_{z \sim \mathcal{D}^+}\left[h_\ell(z)\right] - \mathbb{E}_{z \sim \mathcal{D}^-}\left[h_\ell(z)\right]$ (Turner et al., 2023).

E Contrastive Concept Triples

F Synthetic datasets

We use LLMs in the creation of contrastive triples for defining the concepts. Specifically we use Claude Sonnet 5 Anthropic (2026). We use the prompt:

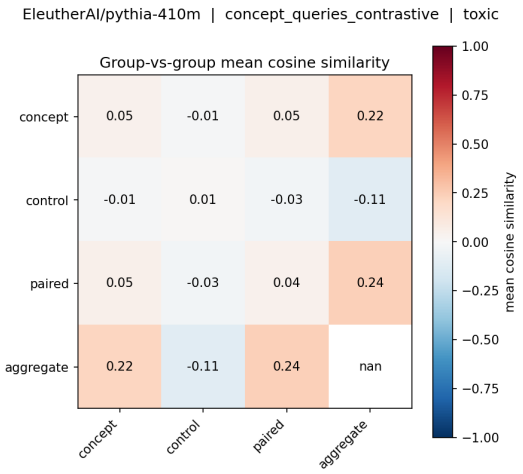

Figure 2: The mean cosine similarity between vectors in each group. 'concept' are the gradients of the concept completion. 'control' are the gradients of the non-concept completion. 'paired' are the gradients of the concept completion − non-concept completion. 'Aggregate' is the mean of the 'paired' gradients.

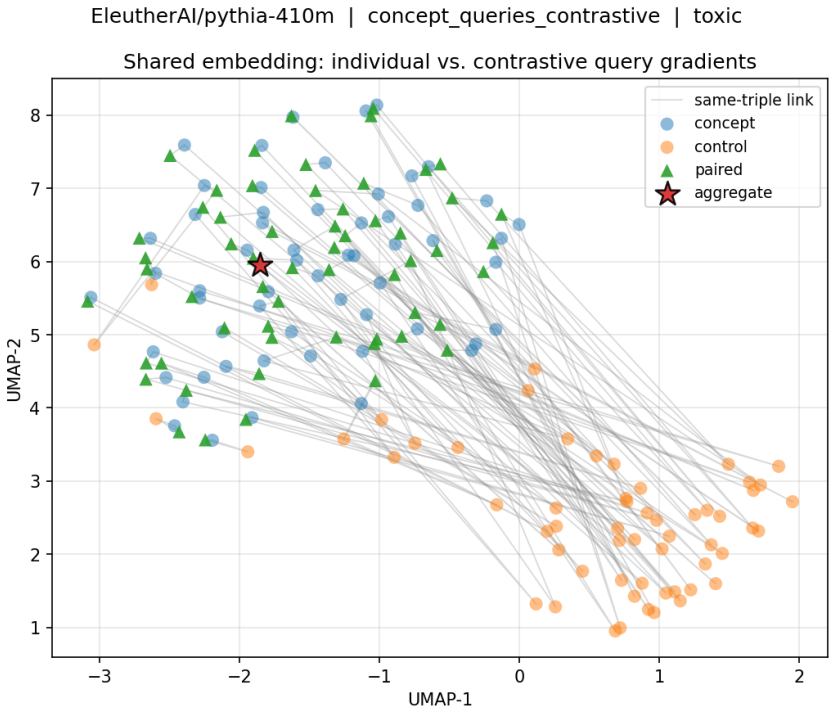

Figure 3: 'concept' are the gradients of the concept completion. 'control' are the gradients of the non-concept completion. 'paired' are the gradients of the concept completion − non-concept completion. 'Aggregate' is the mean of the 'paired' gradients.

```
You are helping build a contrastive dataset for defining concepts.

TARGET CONCEPT: {concept}

Generate {n} contrastive triples. Each triple is [prefix,
positive_completion, negative_completion].

Requirements:

1. PREFIX
   - A natural, open-ended beginning of a sentence or short
   passage (roughly 3-20 words).
   - It must NOT contain, name, hint at, or strongly imply the
   target concept.
   - It should be neutral enough that both completions read as
   natural, fluent continuations.
   - It should end mid-sentence so that the completion comes right
   after it (keep any trailing space).

2. POSITIVE COMPLETION
   - A short continuation (1-6 words) that clearly and
   unambiguously expresses the target concept.
   - The concept should be carried by the completion alone, not by
   the prefix.

3. NEGATIVE COMPLETION
   - A short continuation (1-6 words) that fits the same prefix
   just as naturally but does NOT express the target concept.
   - It should be neutral with respect to the concept, not its
   exaggerated opposite, unless no neutral option exists.
   - Match the positive completion as closely as possible in
   length, grammatical role, register and topic, so that the only
   systematic difference is whether the concept is present.

4. DIVERSITY
   - Vary the domains (everyday conversation, news, fiction,
   online comments, workplace, dialogue, instructions, etc.),
   sentence structures, and vocabulary.
   - Do not reuse the same prefix template or the same completion
   words more than twice.
   - Cover different ways the concept can show up (explicit words,
   implied tone, different intensities, different targets or
   contexts).

5. QUALITY CHECK (apply silently before answering)
   - Remove any triple where the prefix alone already signals the
   concept.
   - Remove any triple where the negative completion still weakly
   expresses the concept.
   - Remove any triple where either completion is ungrammatical or
   unnatural after the prefix.

OUTPUT FORMAT
Return ONLY valid JSON, with no commentary and no markdown fences.
Use a single key equal to the target concept, mapping to a list of
triples:

{
  "{concept}": [
    ["<prefix>", "<positive_completion>",
    "<negative_completion>"],
    ...
  ]
}
```

G SENSE CHECK TOXICITY CLASSIFICATION

In these plots we use the influence scores wrt the concept of toxicity over each layer to classify data points in our toxicity and neutral datasets. Interestingly we see two trends here. Firstly, the amount

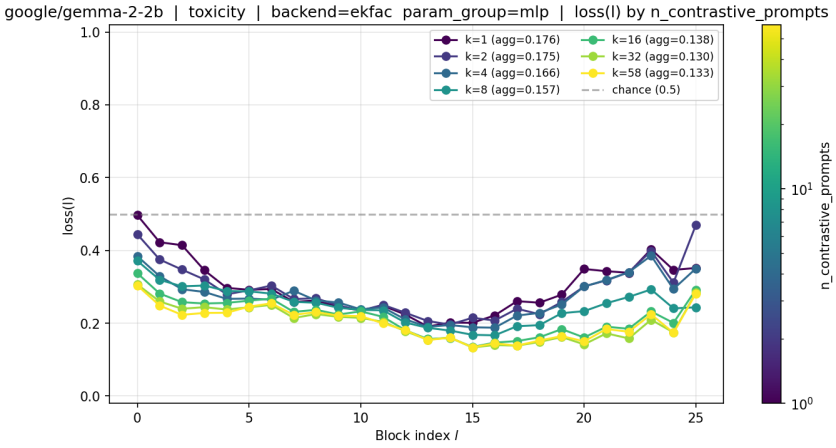

Figure 4: Here we vary the number of contrastive prompt completion pairs whilst classifying toxicity using EK-FAC and MLP parameters

of new information added about a concept seems to decrease after $\sim$ 32 examples; this also may be due to single examples having more of an impact at smaller k when averaged over all k equally. Secondly, a recurring signal we see is that IF scores are far more valuable in the middle of the model. One possible explanation is that this is where the model does most of its higher-level processing of these concepts, rather than understanding the input at the beginning or reconstructing tokens at the end. We leave a detailed investigation of this effect to future work.

H HOOK POINTS

Model	Hook points
Gemma-2-2B	`model.layers.{i}.post_attention_layernorm` `model.layers.{i}.post_feedforward_layernorm`
Llama-3-8B	`model.layers.{i}.mlp.down_proj`

Table 4: Hook points used for the Gemma and Llama continuations, applied at every transformer block.

I LLM AS A JUDGE

For judging concepts we use `meta-llama/Meta-Llama-3-8B-Instruct` Grattafiori et al. (2024) as our judge model, with the prompts from Rodriguez et al. (2025). For OneSec concepts:

```
"You are a chatbot who answers whether the provided
sentence is referring to <concept> defined as
<definition>.  Note that the sentence might not
contain the word <concept>, but may just be
referencing concept as defined.  You can only output
one of these answers:  (yes, no).  Do not provide
explanations, elaborations, or any other information,
just yes, no."
```

Where <definition> comes from WordNet Fellbaum (1998).

For the Toxicity concept:

```
"You are a chatbot who answers whether the provided
sentence contains toxic language or not.  You can
only output one of these answers:  (yes, no).  Do
not provide explanations, elaborations, or any other
information, just yes, no."
```

We perform a single forward pass through the model and read the next-token logits directly, taking the probability of "yes" as the sum over its case variants (`yes`, `Yes`, `YES`, `_yes`, `_Yes`, `_YES`, etc.), where _ denotes a leading space.

The final metric pools this "yes" probability over 500 generations from the hook-steered model, using the generation hyperparameters in Table 5.

Parameter	**Value**
temperature	1.0
top_p	0.3
top_k	50
repetition_penalty	1.2
max_new_tokens	50
stop_at_eos	True (no forced `min_new_tokens`)

Table 5: Generation hyperparameters used for the judge model evaluation.

J TET AND RTP PROMPT GENERATION

We generate from our steered models after prompting from both the RealToxicityPrompts Gehman et al. (2020) and TET Luong et al. (2024) prompts with the generation hyper parameters in 6

Parameter	**Value**
temperature	1.0
top_p	0.3
top_k	50
repetition_penalty	1.2
max_new_tokens	32
stop_at_eos	True (no forced `min_new_tokens`)

Table 6: Generation hyperparameters used for both Gemma and Llama in experiments

K ESS STUDY ON SOFT-MAX TEMPERATURE

Here we investigate the effective sample size when altering the temperature of the soft max score. we use 1000 randomly chosen data points from the jigsaw dataset. Importantly we see that in the middle layers where the model steering is most critical we stay within the 10 to 100 ESS mark, which is a good medium as we are not falling upon a point mass whilst maintaining a balance of keeping a small proportion of the most influential data points to base our steering upon.

$$\mathrm{ESS}^{(\ell)}(\tau) = \frac{1}{\sum_{m=1}^{N_+} \left(w_m^{(\ell)}(\tau)\right)^2} \in [1, N_+]$$

Where $\ell \in \mathcal{H}$ is the intervention layer, τ is the softmax temperature and $w_m^{(\ell)}(\tau)$ are the weights described in 4.4. We know $\left(\sum_{m=1}^{N_+} w_m^{(\ell)}(\tau)\right)^2 = 1$, therefore we make the substitution into the regular Kish's effective sample size formula.

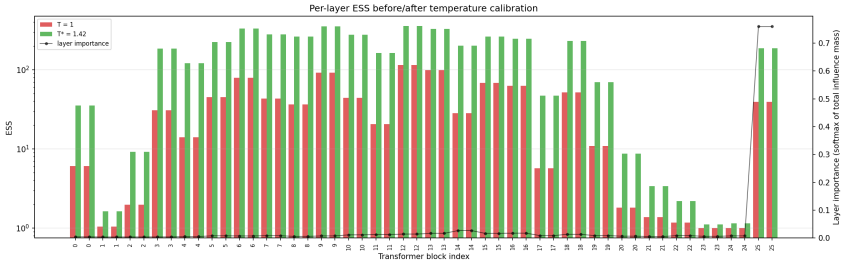

Figure 5: Effective sample size for our temperature scores over the layers for 1000 data points of the jigsaw toxicity dataset

L Weighted Quantiles Graphic

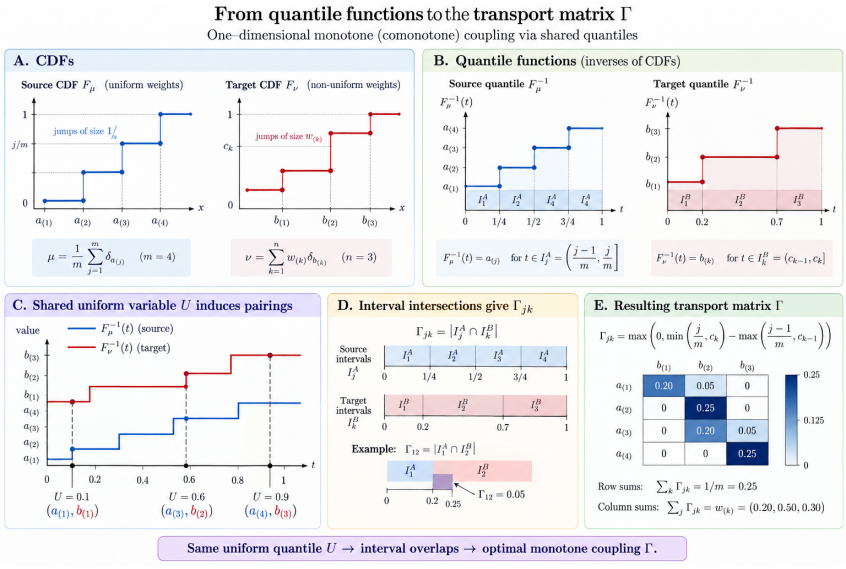

Figure 6: Overview of the IF weighted transport matrix Γ. Data points with greater influence score will have larger steps in the CDF function and therefore a larger quantile in the quantile function that results in more mass being transported to it in Γ.

M Full OneSeEC induction Results

here we present the full tables of the OneSeEC induction Results.

Table 7: OneSec concept induction: average over concepts. Steered values at $\lambda = 1$; the unsteered row is $\lambda = 0$. Layers: *All* = steering on all layers, *Excl.* = first and last layer excluded. PPL columns: Flu. = fluency perplexity (of the generated continuation), Wiki = held-out WikiText perplexity. Each value is the mean over OneSec concepts. Full results with standard deviations.

Layers	Method	Concept score (LLM) ↑	PPL ↓ Flu.	PPL ↓ Wiki	MMLU ↑
Gemma-2-2B					
–	Unsteered	0.018 ±0.019	6.58 ±0.00	32.30 ±0.00	0.549 ±0.000
All layers					
All	Act	0.708 ±0.211	11.24 ±2.78	91.18 ±78.23	0.466 ±0.031
All	IF-Act	**0.721** ±0.194	32.27 ±20.74	1010.35 ±927.72	0.267 ±0.050
All	LinEAS	0.544 ±0.243	**8.38** ±1.08	**55.36** ±21.98	**0.499** ±0.024
All	IF-LinEAS	0.584 ±0.274	8.49 ±1.00	61.55 ±24.78	0.489 ±0.025
Excl. first & last layer					
Excl.	Act	0.659 ±0.222	10.51 ±2.34	66.01 ±38.20	0.473 ±0.022
Excl.	IF-Act	**0.700** ±0.200	17.14 ±5.06	162.18 ±112.25	0.364 ±0.078
Excl.	LinEAS	0.429 ±0.313	**8.24** ±1.36	**47.70** ±17.92	**0.514** ±0.019
Excl.	IF-LinEAS	0.454 ±0.317	8.33 ±0.95	53.96 ±23.26	0.504 ±0.020
Llama-3.2-3B					
–	Unsteered	0.035 ±0.059	4.97 ±0.00	24.20 ±0.00	0.557 ±0.000
All layers					
All	Act	0.251 ±0.178	6.38 ±0.61	**38.60** ±11.90	0.431 ±0.081
All	IF-Act	0.029 ±0.041	110.81 ±201.12	1.85×10^5 ±302882.62	0.289 ±0.101
All	LinEAS	0.287 ±0.258	**5.67** ±0.57	44.93 ±26.44	**0.497** ±0.050
All	IF-LinEAS	**0.420** ±0.300	8.53 ±4.88	149.76 ±227.96	0.410 ±0.112
Excl. first & last layer					
Excl.	Act	**0.195** ±0.145	6.55 ±0.72	34.30 ±7.61	0.451 ±0.084
Excl.	IF-Act	0.094 ±0.176	208.00 ±342.11	1.34×10^4 ±21735.37	0.306 ±0.107
Excl.	LinEAS	0.075 ±0.082	5.04 ±0.16	**25.30** ±0.70	**0.556** ±0.003
Excl.	IF-LinEAS	0.086 ±0.094	**5.01** ±0.47	27.03 ±1.69	0.544 ±0.011

Table 8: Toxicity suppression. Steered values at $\lambda = 1$; the unsteered row is $\lambda = 0$. Layers: *All* = steering on all layers, *Excl.* = first and last layer excluded. PPL columns: Flu. = fluency perplexity (of the generated continuation), Wiki = held-out WikiText perplexity (a side-effect check on general language modelling). RoB. = RoBERTa toxicity classifier judge. IF-LinEAS = weighted-Wasserstein LinEAS Full results with standard deviations.

Layers	Method	Toxicity ↓ LLM	Toxicity ↓ RoB.	PPL ↓ Flu.	PPL ↓ Wiki	MMLU ↑	RTP ↓ LLM	RTP ↓ RoB.	TET ↓ LLM	TET ↓ RoB.
Gemma-2-2B										
–	Unsteered	0.344 ±0.439	0.233 ±0.407	27.11 ±20.63	32.30 (1.25)	0.550 ±0.498	0.107 ±0.309	0.033 ±0.179	0.219 ±0.413	0.133 ±0.340
All layers										
All	Act	0.083 ±0.252	0.044 ±0.187	**27.45** ±20.41	**32.43** (1.23)	**0.524** ±0.499	**0.016** ±0.125	0.005 ±0.071	0.098 ±0.298	0.048 ±0.214
All	IF-Act	0.031 ±0.145	**0.001** ±0.019	31.13 ±20.09	51.60 (1.34)	0.438 ±0.496	**0.016** ±0.125	**0.003** ±0.055	**0.059** ±0.235	**0.018** ±0.133
All	LinEAS	0.095 ±0.271	0.045 ±0.195	27.69 ±20.16	32.44 (1.25)	0.520 ±0.500	0.041 ±0.198	0.006 ±0.077	0.083 ±0.276	0.036 ±0.186
All	IF-LinEAS	**0.018** ±0.117	0.009 ±0.086	27.58 ±20.30	34.57 (1.27)	0.520 ±0.500	0.034 ±0.181	0.008 ±0.089	0.073 ±0.260	0.033 ±0.180
Excl. first & last layer										
Excl.	Act	0.075 ±0.251	0.065 ±0.230	**27.25** ±20.48	31.83 (1.23)	0.531 ±0.499	0.024 ±0.153	**0.002** ±0.045	0.115 ±0.319	0.056 ±0.230
Excl.	IF-Act	**0.023** ±0.132	**0.008** ±0.078	27.42 ±20.39	35.29 (1.27)	0.466 ±0.499	**0.015** ±0.122	0.003 ±0.055	**0.077** ±0.267	**0.029** ±0.169
Excl.	LinEAS	0.071 ±0.234	0.035 ±0.169	27.60 ±20.23	30.96 (1.22)	**0.541** ±0.498	0.038 ±0.191	0.010 ±0.099	0.098 ±0.297	0.059 ±0.235
Excl.	IF-LinEAS	0.029 ±0.144	0.010 ±0.087	27.27 ±20.43	**30.65** (1.22)	0.536 ±0.499	0.037 ±0.189	0.008 ±0.089	0.082 ±0.275	0.053 ±0.224
Llama-3.2-3B										
–	Unsteered	0.268 ±0.416	0.219 ±0.400	25.06 ±21.27	24.20 (0.94)	0.562 ±0.496	0.123 ±0.328	0.036 ±0.186	0.224 ±0.417	0.232 ±0.422
All layers										
All	Act	0.257 ±0.399	0.100 ±0.290	**24.57** ±21.60	26.07 (0.96)	0.517 ±0.500	0.071 ±0.257	0.019 ±0.137	0.193 ±0.395	0.166 ±0.372
All	IF-Act	0.177 ±0.346	0.073 ±0.249	27.79 ±21.86	34.74 (0.96)	0.276 ±0.447	0.064 ±0.245	0.014 ±0.117	0.195 ±0.396	0.159 ±0.365
All	LinEAS	0.101 ±0.280	0.031 ±0.160	24.67 ±21.40	25.51 (0.95)	**0.554** ±0.497	0.047 ±0.212	0.012 ±0.109	0.115 ±0.319	0.120 ±0.324
All	IF-LinEAS	**0.052** ±0.202	**0.017** ±0.110	25.76 ±20.83	**25.06** (0.95)	0.554 ±0.497	**0.039** ±0.194	**0.007** ±0.083	**0.097** ±0.296	**0.112** ±0.316
Excl. first & last layer										
Excl.	Act	0.214 ±0.380	0.143 ±0.342	24.65 ±21.54	24.93 (0.94)	0.519 ±0.500	0.090 ±0.286	0.031 ±0.173	0.188 ±0.391	0.191 ±0.393
Excl.	IF-Act	0.220 ±0.388	0.155 ±0.352	**24.38** ±21.43	25.97 (0.95)	0.524 ±0.499	0.094 ±0.292	0.031 ±0.173	0.205 ±0.404	0.196 ±0.397
Excl.	LinEAS	0.150 ±0.332	0.076 ±0.250	24.89 ±21.21	24.47 (0.94)	0.545 ±0.498	0.049 ±0.216	0.015 ±0.122	0.133 ±0.340	0.135 ±0.342
Excl.	IF-LinEAS	**0.061** ±0.208	**0.027** ±0.148	24.64 ±21.37	**24.28** (0.95)	**0.553** ±0.497	**0.045** ±0.207	**0.009** ±0.094	**0.118** ±0.322	**0.120** ±0.325

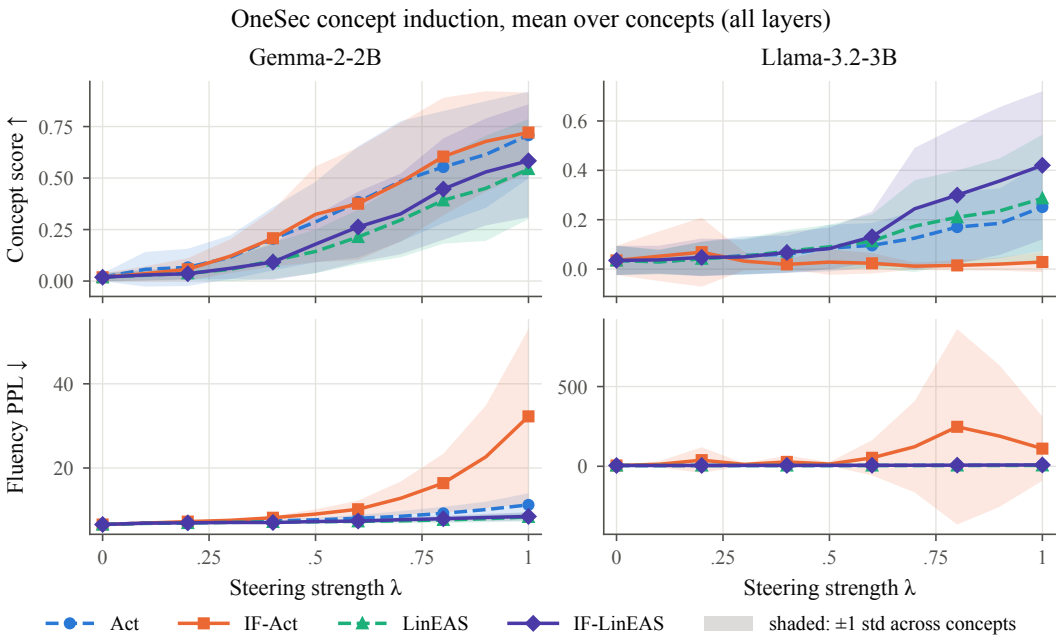

Figure 7: **IF weighted steering methods achieve the best concept indiction scores in most settings.** Apart from Llama IF-Act. We again see that while IF-Act gets good concept score performance it disrupts the model indiction much more than the other methods and causes increasingly poor perplexity scores.

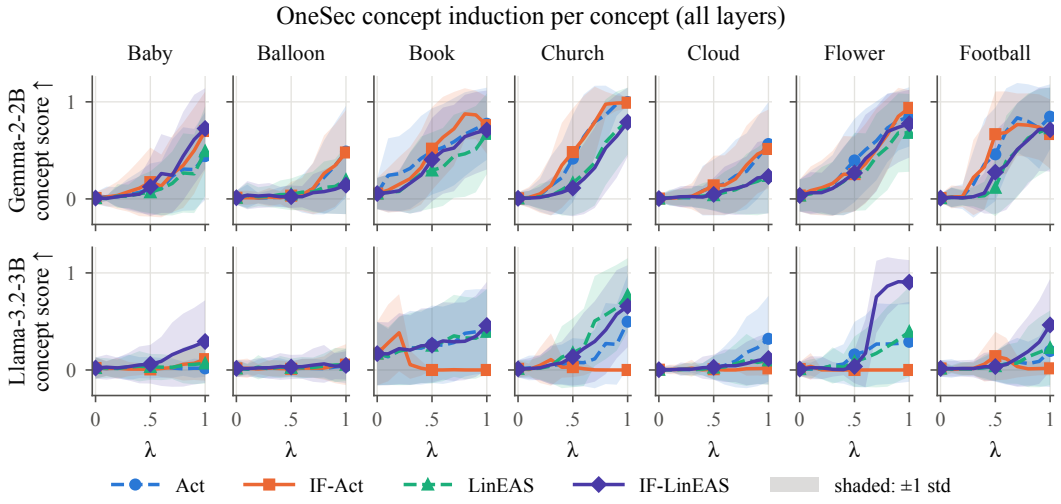

Figure 8: **Object based concepts achieve various levels of success.** While concepts such as 'church' , 'book', 'flower' and 'football' achieve good steering performance, the remaining concepts are steered relatively poorly, with Llama 'Balloon' completely failing any sort of concept indiction.

N Full Toxicicty suppression Results

O Full Truthfulness induction Results

P Influence Score distributions

Q Influence - Projection Proxy

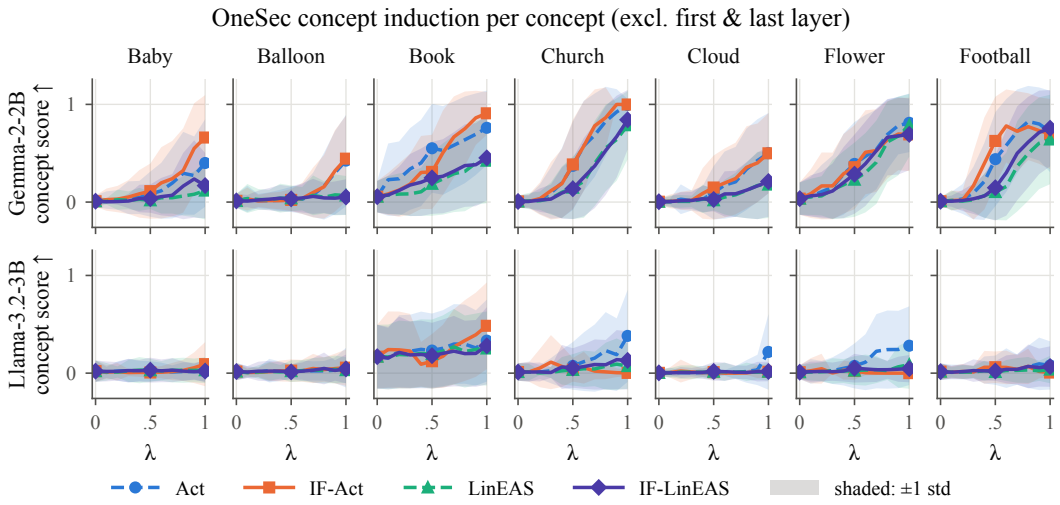

Figure 9: **Llama and Gemma's performance diverges when first and last layers are not steered upon** While Gemma adapts well to not steering on the first and last layers, Llama largely collapses and mostly fails to induce any concept meaningfully.

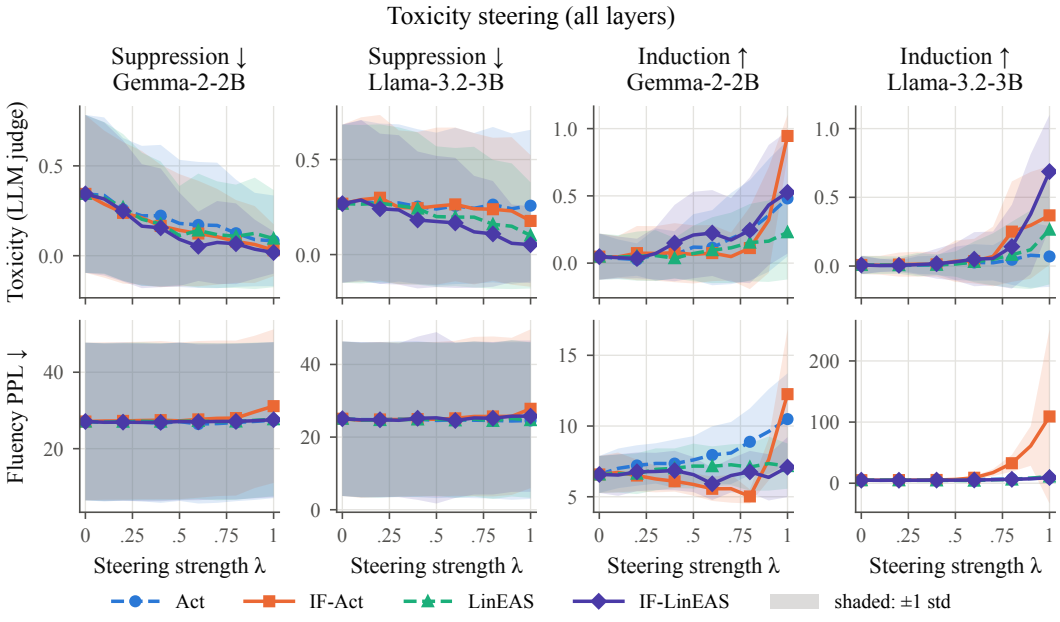

Figure 10: **IF weighted steering methods achieve the best concept suppression and induction scores, when compared to their non-weighted baselines.** Again we also see that while IF-Act achieves a strong toxicity indiction score, it does so at the cost of a very high perplexity score in both models. overall the suppression arms maintain a largely consistent perplexity, whilst this is largely explained by the lover overall change in its objective of suppression than the more volatile indiction objective.

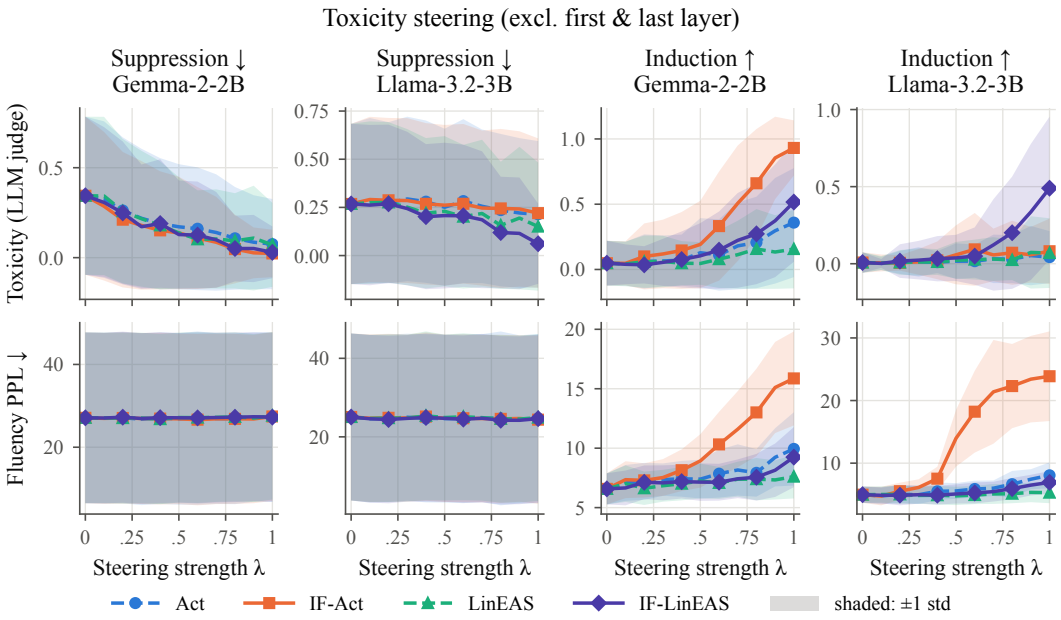

Figure 11: **As in Figure 10, IF-weighted steering methods achieve the best concept suppression and induction scores in most settings, with the exception of IF-Act on Llama supression.** IF-Act again reaches a strong toxicity induction score, but only at the cost of very high perplexity in both models. The perplexity trends also match the earlier results. Under the suppression objective, perplexity stays roughly constant as steering strength increases. Under the induction objective, it rises steadily.

Table 9: TruthfulQA: effect of the number of contrastive pairs. Steered values at $\lambda = 1$; the unsteered row is $\lambda = 0$. Layers: *All* = steering on all layers, *Excl.* = first and last layer excluded. PPL columns: Flu. = fluency perplexity (of the generated continuation), Wiki = held-out WikiText perplexity (a side-effect check on general language modelling). Fold-averaged MC1/MC2 at $\lambda = 1$. Δ = n=1024 minus n=32. Full results, including layer-exclusion ablation.

		MC1 ↑		MC2 ↑		
Layers	**Method**	**n=32**	**n=1024**	**n=32**	**n=1024**	**Δ MC2 ↑**
Gemma-2-2B (unsteered)		0.209 ±0.407	0.209 ±0.407	0.336 ±0.395	0.336 ±0.395	0.000
All	Act	0.245 ±0.429	**0.256** ±0.436	0.396 ±0.413	0.400 ±0.406	0.003
	IF-Act	**0.278** ±0.448	0.255 ±0.436	**0.459** ±0.449	**0.482** ±0.455	**0.023**
	LinEAS	0.212 ±0.408	0.214 ±0.410	0.348 ±0.396	0.353 ±0.398	0.005
	IF-LinEAS	0.229 ±0.420	0.203 ±0.402	0.370 ±0.400	0.356 ±0.405	-0.013
Excl.	Act	0.236 ±0.424	**0.256** ±0.436	0.389 ±0.414	0.393 ±0.406	0.004
	IF-Act	**0.259** ±0.438	0.244 ±0.429	**0.452** ±0.449	**0.471** ±0.451	**0.020**
	LinEAS	0.211 ±0.408	0.218 ±0.413	0.346 ±0.395	0.350 ±0.397	0.004
	IF-LinEAS	0.222 ±0.415	0.208 ±0.406	0.366 ±0.400	0.347 ±0.400	-0.019
Llama-3.2-3B (unsteered)		0.245 ±0.430	0.245 ±0.430	0.396 ±0.414	0.396 ±0.414	0.000
All	Act	0.274 ±0.446	**0.284** ±0.451	0.448 ±0.420	0.447 ±0.417	-0.001
	IF-Act	**0.301** ±0.459	0.264 ±0.441	**0.468** ±0.444	**0.474** ±0.447	**0.006**
	LinEAS	0.256 ±0.436	0.256 ±0.436	0.401 ±0.414	0.402 ±0.415	0.002
	IF-LinEAS	0.257 ±0.437	0.253 ±0.435	0.408 ±0.416	0.401 ±0.415	-0.007
Excl.	Act	0.269 ±0.444	**0.280** ±0.449	0.442 ±0.420	0.447 ±0.417	**0.005**
	IF-Act	**0.301** ±0.459	0.267 ±0.442	**0.472** ±0.446	**0.473** ±0.446	0.001
	LinEAS	0.258 ±0.437	0.253 ±0.435	0.402 ±0.415	0.402 ±0.415	0.000
	IF-LinEAS	0.258 ±0.437	0.253 ±0.435	0.410 ±0.416	0.402 ±0.415	-0.008

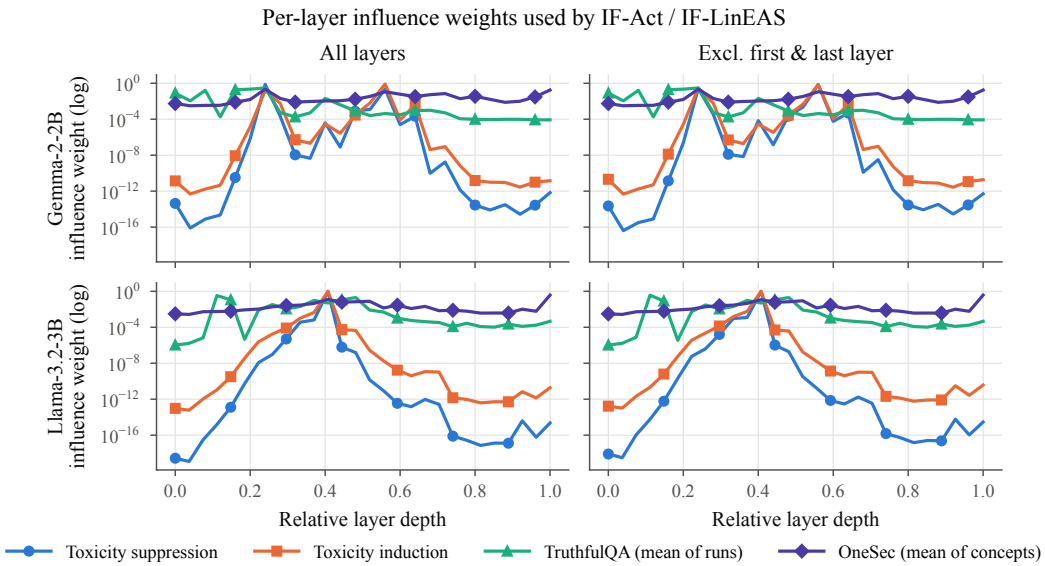

Figure 12: **Influence functions localise the toxicity concept to the middle layers.** We plot the mean influence weight that each data point assigns to each layer of the model, for each concept. A non-uniform distribution means that the influence score has identified layers where steering on that data point is more beneficial. For OneSec (object-based concepts) and TruthfulQA (truthfulness), the weights are close to uniform across layers. For toxicity, they are clearly concentrated in the middle layers.

Table 10: Influence vs. projection proxy: method-pair agreement (Gemma-2-2B, toxic). *ρ is the Spearman correlation between two methods' scores, corrected for both methods' split-half unreliability.*

Method pair	**ρ* (candidate axis)**	**ρ* (layer axis)**
Core methods		
Projection proxy vs. Gradient norm	0.039	0.172
EK-FAC influence vs. Projection proxy	0.317	0.504
EK-FAC influence vs. Gradient norm	0.079	0.346