DISTRIBUTED SPARSE INTERVENTIONS IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models can perform a wide range of tasks at various levels of abstraction. They flexibly infer tasks from context, execute multiple tasks simultaneously, and select among competing goals. Understanding and controlling how models represent these tasks is central to aligning them with human objectives. Researchers can change model behaviour in many ways, but model steering, which manipulates intermediate task representations, links representation to behaviour most directly. Current steering methods typically intervene densely and assume linear effects, e.g., shifting every activation along a global direction in the residual stream. Because such interventions change thousands of neurons at once, they leave researchers to separate relevant from irrelevant neurons. We introduce Distributed Sparse Interventions (DSI), which steers models through a sparse set of individual neurons distributed across layers and attention heads. Finding such a set requires accounting for the selected neurons' interactions, even across layers; DSI achieves that by iteratively refining which neurons to intervene on and how strongly. Across 12 tasks and three instruction-tuned models, DSI activates task behaviour by identifying and intervening on fewer than 0.1% of neurons, outperforming dense steering vectors and often matching 10-shot in-context learning. Because DSI identifies explicit sets of neurons, it connects neuron-level computation to changes in latent representations and, ultimately, to desired outputs, allowing us to study how distributed computational components interact to produce steered behaviour. Our results demonstrate that steering can be viewed not only as a linear shift in representation space, but also as a targeted intervention on model computation, realised by adjusting a small set of interacting neurons whose coordinated activity spans multiple layers and contributes to behaviour.

1 INTRODUCTION

Modern language models exhibit flexible task-solving capabilities; through in-context learning (ICL), they infer tasks from context, execute multiple tasks simultaneously, and select among competing goals (Brown et al., 2020; Li et al., 2024a; Xiong et al., 2025). Understanding and controlling how models represent these tasks is essential to safeguard models and ensure alignment with human objectives. Researchers can change model behaviour in several ways, most commonly through prompt engineering and fine-tuning. Model steering offers a third approach by intervening on intermediate representations that modify the model's forward pass to activate or suppress specific behaviours. Because steering manipulates task representations, it links representation to behaviour most directly, making it well suited for both flexible, context-dependent adaptation and studying how tasks are represented. Current steering methods typically rest on three assumptions. First, they assume that intervention effects are approximately linear: most methods identify directions in activation space (Rimsky et al., 2024; Todd et al., 2024; Hendel et al., 2023; Beaglehole et al., 2026; Liu et al., 2024; Wu et al., 2025a), leveraging the observation that many task-relevant behaviours exhibit approximately linear structure. Second, they assume that interventions are dense: whether a vector is estimated directly at a specific layer (Rimsky et al., 2024; Hendel et al., 2023) or composed from sparse dictionary features (Templeton et al., 2024), the resulting intervention modifies large numbers of neurons simultaneously, since features that are sparse in a learned dictionary are dense in the neuron basis (Bricken et al., 2023; Elhage et al., 2022). Recent work has begun to restrict interventions to selected attention heads (Yin et al., 2024) or subsets of neurons (Feng et al., 2026), showing that sparser interventions can be effective. Third, interactions between components are often

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.

left out: many methods intervene at a single, predefined layer (Rimsky et al., 2024; Todd et al., 2024), and methods that optimise interventions typically capture interactions only within that layer or include whole attention heads (Wu et al., 2025a;b; Yin et al., 2024). These assumptions limit analysis to whole directions or components, leaving researchers to separate relevant from irrelevant neurons. We introduce *Distributed Sparse Interventions* (DSI), which relaxes all three assumptions. By iteratively refining which neurons to intervene on and how strongly, DSI captures nonlinear effects and interactions between neurons, while restricting the intervention to a sparse set distributed across layers and attention heads (Figure 1). Our contributions are as follows:

(1) An analysis of neuron-level interventions leading to DSI, a method enabling task activation by identifying and intervening on sparse sets of neurons (Section 3).

(2) A demonstration of complex, non-linear dependencies among neuron interventions, highlighting the importance of iterative approaches to determine multi-layer interventions (Section 4).

(3) Across 12 tasks (Todd et al., 2024) and three models (Qwen3 (8B) (Qwen Team, 2025), Gemma 3 (4B) (Gemma Team, 2025), Llama 3.2 (3B) (Grattafiori et al., 2024)), intervening on 1–64 neurons (fewer than 0.1% of attention-head neurons) typically suffices to activate a task, outperforming dense function vectors (Todd et al., 2024) and often matching 10-shot ICL (Section 4.1, Table 1; Section 4.2), Figure 3.

(4) At small budgets, interventions distributed across layers clearly outperform interventions restricted to a single layer (Section 4.3, Figure 4).

(5) On AxBench (Wu et al., 2025a), a benchmark for steering several hundred concepts in open-ended text generation, DSI performs on par with recent steering methods at larger budgets, and clustering its interventions shows that similar neuron sets steer semantically related concepts (Section 5, Figure 5).

Together, these results challenge the assumption that steering requires dense, linear interventions at specific layers, and show the potential of sparse neuron sets as a more interpretable unit for the study of task representations. Our code will be made openly available upon publication.

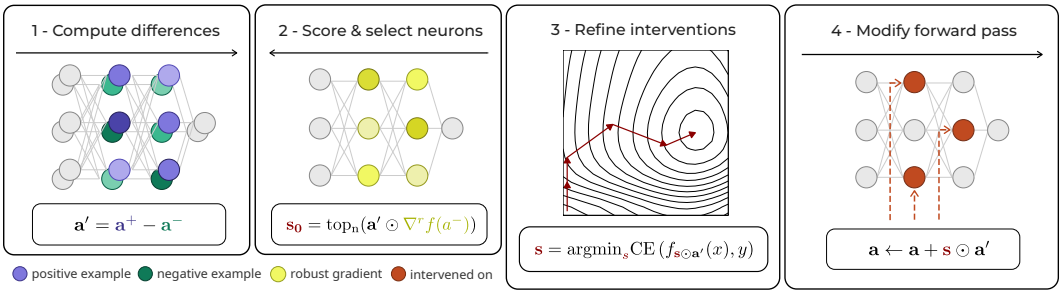


Figure 1: Schematic overview of how distributed sparse interventions are identified and applied via DSI. (1) Activation differences are extracted from positive and negative examples. (2) An initial sparse intervention is computed using a robustified gradient estimate. (3) The intervention is refined via an iterative procedure. (4) The refined intervention is applied additively to modify model behaviour.

2 Related Work

Model steering Prior work on model steering has largely focused on identifying linear directions in activation space to intervene on model behaviour. Associated vectors are typically calculated from in-context-learning examples (Todd et al., 2024; Hendel et al., 2023; Liu et al., 2024) as well as contrastive sample pairs (Rimsky et al., 2024). More recently, methods have been proposed that identify vectors via optimisation, such as ReFT-r1 (Wu et al., 2025a), combining concept detection and steering, or RePS (Wu et al., 2025b), proposing a novel preference-optimisation objective. Moving towards sparse interventions, LoFiT (Yin et al., 2024) trains additive biases applied to specific attention heads, while Feng et al. (2026) show that restricting interventions to a subset of neurons can increase steering performance. Moreover, conditional interventions have been explored in works such as HyperSteer (Sun et al., 2025), a hypernetwork-based approach computing steering

vectors dependent on the steered model's representations. In contrast, we propose to intervene on a sparse set of neurons, modelling their nonlinearities and interactions across attention heads and layers.

Sparse representations in neural networks Often inspired by neuroscientific findings of sparse coding (Vargas-Barroso et al., 2026; Olshausen & Field, 2004; Olshausen et al., 2003), prior work on understanding representations in artificial neural networks has investigated sparse representations. Research on model compression and pruning (Li et al., 2024b), as well as the lottery ticket hypothesis (Frankle & Carbin, 2019; Evci et al., 2020), suggests the existence of underlying sparse structure in large deep learning models by identifying sparse subnetworks capable of matching the performance of dense models. In vision models, neuron units aligning with human-interpretable concepts were identified (Bau et al., 2017). Similarly, in language models, specific 'skills' (Wang et al., 2022) and 'knowledge' (Dai et al., 2022) have been localised to neurons, and sparse subnetworks have been shown to support strong performance through targeted fine-tuning (Panigrahi et al., 2023). As neural models may represent features in superposition (Elhage et al., 2022), recent interpretability work has explored dictionary learning to disentangle representations into interpretable features (Bricken et al., 2023), assuming the native neuron basis is not sufficiently sparse. Further efforts have focused on learning a sparse basis of interpretable features either at a specific layer (e.g., sparse autoencoders (Bricken et al., 2023; Huben et al., 2023)) or across multiple layers (e.g., crosscoders (Lindsey et al., 2024)) in order to generate sparser circuits (Marks et al., 2025). While this line of work relies on dense features in high-dimensional spaces, it remains an open question to what extent task-relevant structure is accessible within the native neuron basis. More recently, it has been found that for some circuits previously analysed with sparse autoencoders, circuits of similar sparsity can be recovered in the neuron basis of the MLP components of transformer blocks (Arora et al., 2026).

3 DSI - Distributed Sparse Interventions

We introduce *Distributed Sparse Interventions* (DSI), a method to identify and apply effective interventions on small sets of neurons distributed across layers and attention heads. As illustrated in Figure 1, DSI follows a four-step procedure: (1) **activation differences** between positive and negative examples are computed for all neurons to isolate task-specific model responses; (2) a sparse **initial intervention** is derived from these activation differences and a robustified gradient; (3) a **refined intervention** is generated via step-wise optimisation of the sparse neuron set and neuron-level intervention scaling, accounting for neuron interactions (Appendix A provides an illustration and analysis demonstrating the necessity of modeling these interactions); and (4) during a **modified forward pass**, this intervention is applied to elicit the task-solving capabilities on 0-shot examples.

After discussing preliminaries, first-order intervention effect estimation and robustified gradients are defined in Section 3.1, and the iterative intervention refinement procedure is introduced in Section 3.2.

Notation Let f denote a model mapping a sequence x of input tokens to output logits. A task is defined via a set D of input-output pairs (x, y). From D, we construct positive examples $D^+ = \{(x^+, y^+)\}$ where the model shows the desired behaviour, e.g., via prompting it to solve a certain task or providing in-context examples. Similarly, we construct negative examples $D^- = \{(x^-, y^-)\}$ where a model does not show the desired behaviour, e.g., 0-shot examples or prompts unrelated to the given task. For an input x, $f(x)_{y^+}$ refers to the output logit of the correct token, i.e., the first token of y^+.

We let $a \in \mathbb{R}$ denote a neuron activation in a hidden layer of f and $\delta \in \mathbb{R}$ an intervention on that neuron. $f_\delta \coloneqq f(x|a \leftarrow a + \delta)$ then denotes the model modified by intervention δ, that is, during a forward pass, a is computed from the input, then δ is added to a, and the forward pass is continued. $\mathbf{a} \in \mathbb{R}^d$ denotes the vector of all hidden activations that are considered for an intervention, and $\Delta \in \mathbb{R}^d$ the vector of all interventions.

3.1 Approximating intervention effects

Taylor approximation To identify which neurons to intervene on for task D, we consider the effect of intervening on a neuron during a forward pass in the negative setting. That is, for a prompt x^-, the effect of adding an intervention δ to a neuron a during the forward pass is

$$\mathcal{E}(\delta) = f(x^-|a \leftarrow a + \delta)_{y^+} - f(x^-)_{y^+}, \tag{1}$$

i.e., the change in the correct output logit caused by the addition of δ to a during the forward pass. Since an exact calculation would require an independent forward pass for each neuron in f, we approximate the intervention effect via a first-order Taylor approximation:

$$f(x^-|a \leftarrow a + \delta)_{y^+} \approx f(x^-)_{y^+} + \partial_a f(x^-)_{y^+}\ \delta$$
$$\implies \mathcal{E}(\delta) \approx \partial_a f(x^-)_{y^+}\ \delta$$

This is similar to the setting of attribution patching (Nanda, 2023), with the difference that the activation of a is not replaced during the forward pass (a "structural" intervention) but a term δ is added to it (a "parametric" intervention) (Eberhardt & Scheines, 2007).

Robustified gradient Deep neural network gradients suffer from instabilities such as gradient shattering (Balduzzi et al., 2017), making them a potentially unreliable signal for the expected changes in model output under interventions. To mitigate this brittle behaviour, various other methods, such as integrated gradients (Sundararajan et al., 2017), smooth grad (Smilkov et al., 2017), or layer-wise relevance propagation (LRP) (Bach et al., 2015; Montavon et al., 2019) have been proposed for computing more robust input attributions. Recent work extends these findings to activation patching and shows that intermediate attributions, using LRP-modified gradients, better estimate patching effects in transformer-based language models than raw gradients (Jafari et al., 2025). To improve the approximation of the intervention effect, we therefore use an LRP-modified gradient. The modifications include using the identity rule (Rezaei Jafari et al., 2024) for nonlinear activation functions, the half rule for gated MLPs and attention modules (Arras et al., 2022; Achtibat et al., 2024), and the LayerNorm rule for layer normalization modules (Ali et al., 2022). We denote this robustified gradient by ∂^r and provide additional details on the rules in Appendix E.

Average effect Equipped with this robustified gradient, we approximate the average effect $\bar{\mathcal{E}}$ of the intervention δ across the dataset of negative examples as

$$\bar{\mathcal{E}}(\delta) \approx \frac{1}{|D^-|} \sum_{x^- \in D^-} \partial_a^r f(x^-)_{y^+}\ \delta. \tag{2}$$

3.2 Determining the interventions

Initial set To determine the magnitude of interventions, we follow prior approaches that construct steering signals via activation differences (Rimsky et al., 2024; Todd et al., 2024) and compute the average activation difference of neuron a between a positive and a negative setting during the forward pass:

$$a' \coloneqq \frac{1}{|D^+|} \sum_{x^+ \in D^+} (a|x^+) - \frac{1}{|D^-|} \sum_{x^- \in D^-} (a|x^-). \tag{3}$$

The resulting quantity captures how a neuron's activation shifts between settings where the task is solved and settings where it is not, providing an informative initialization for intervention strengths. For a given number of neurons n to intervene on, we then choose the neurons with the largest expected effect $\bar{\mathcal{E}}(a')$ according to Equation 2. Introducing a scaling factor $\mathbf{s} \in \mathbb{R}^d$ which is one for the n largest effects and zero otherwise, we write the vector of interventions as $\Delta = \mathbf{s} \odot \mathbf{a}'$.

Iterative refinement So far, we approximated the expected effect of an intervention δ on a neuron a with a first-order Taylor expansion around the root point $(a|x^-)$, i.e., the value the neuron takes on during a forward pass on a 0-shot prompt. This disregards nonlinear effects of interventions, as well as interaction effects among interventions, which, as we show in Section 4, can be sizeable in highly nonlinear transformer networks. As explicit computation of higher-order terms across thousands of interventions is computationally infeasible, we opt for an iterative approximation. When maximizing a nonlinear function, it is typically not approximated only once at the initial point, but the function is repeatedly approximated, followed by a step in the gradient direction, and this process is iterated until a stopping criterion is met. Appendix A contains an illustrative example showing how such an iterative approach can help model interactions between neuron-level interventions.

By first constructing a candidate set of n neurons with the largest expected effect $\bar{\mathcal{E}}(a')$ (as described above), and iteratively refining the intervention, we arrive at our proposed Algorithm 1: for a given neuron set size n and datasets D^- and D^+, it (1) computes the average activation differences, (2) computes a robustified gradient w.r.t. the activations in the 0-shot setting, (3) selects the n

interventions with the highest expected effect as a starting point and, (4) iteratively refines the effects t-times by re-evaluating the expected effects at the current intervention.

For the iterative optimisation steps, we employ ZeroFPR (Themelis et al., 2018), an algorithm suited for optimising nonconvex functions with a limited number of nonzero parameters (the per-neuron interventions). Instead of maximising the raw output logit corresponding to the correct answer, we minimise the cross-entropy between the predicted probabilities and the correct output $\mathrm{CE}(f(x), y) := \log\left(\sum_i e^{f(x)_i}\right) - f(x)_y$, since it discourages interventions that inflate many logits simultaneously. We set a maximum number of 50 steps and re-run the algorithm with multiple perturbed starting values to encourage exploring different neuron sets. Details about the optimisation procedure can be found in Appendix F. Note that instead of directly estimating the interventions, we estimate $\mathbf{s}$, the scaling factor of the activation differences. This informs the optimiser about the expected changes in the activations, letting $\mathbf{a}'$ act as a preconditioner for the optimisation problem.

Algorithm 1 Compute DSI

Require: dataset D^-, dataset D^+, set size n, number of steps t

$\mathbf{a}' \leftarrow \mathrm{mean}_{x \in D^+}(\mathbf{a}|x) - \mathrm{mean}_{x \in D^-}(\mathbf{a}|x)$ ▷ average activation difference

$\mathbf{g} \leftarrow \mathrm{mean}_{x \in D^-} \nabla^r_{\mathbf{a}} f(x)_{y^+}$ ▷ robustified gradient in 0-shot setting

$\mathbf{e} \leftarrow \mathbf{g} \odot \mathbf{a}'$ ▷ expected first-order effect of interventions

$s_0 \leftarrow \mathbf{1}(\mathrm{topn}(\mathbf{e}))$ ▷ most relevant neurons as starting point

for $i = 1$ **to** t **do**

  $s_i \approx \arg\min_s \mathrm{mean}_{x \in D^-} \mathrm{CE}\left(f_{s \odot \mathbf{a}'}(x), y\right)$ ▷ update intervention

    s.t. $\mathrm{nnz}(s) \leq n,\ s$ close to s_{i-1} ▷ sparse & close to previous step

end for

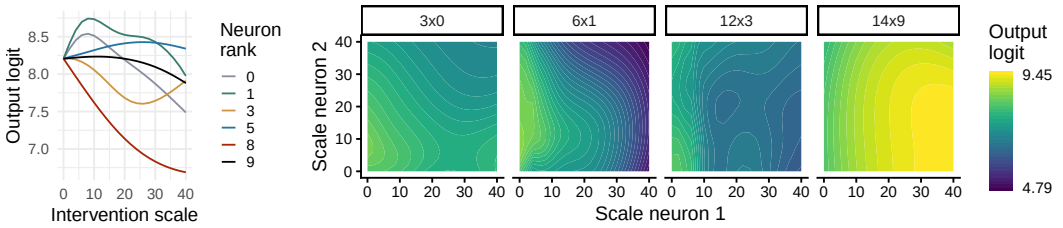
return s_t

4 Experiments

Experimental setup We evaluate our approach on three instruction-tuned language models from widely used families, Qwen 3 (8B) (Qwen Team, 2025), Gemma 3 (4B) (Gemma Team, 2025), and Llama 3.2 (3B) (Grattafiori et al., 2024) (see Appendix C for details), spanning a range of model depths (28-36 layers), attention heads (8-32 heads), and hidden dimensions (128, 256). We consider a subset of 12 tasks from Todd et al. (2024), including two abstraction levels (see Table 3 for an overview): *extractive*, where the correct output is already present in the input (e.g., choosing the fruit from a set of words), and *abstractive*, where the output is not present in the input (e.g., naming the capital of a given country). For each task, we split the input-output pairs 50/50 into a training set and a test set. We then generate 100 0-shot prompts and 100 10-shot prompts from the training split. Two datasets contain fewer samples, resulting in smaller train and test splits. Details on the task data, splits, and prompt structure are provided in Appendix D. For the interventions, we consider the outputs of all attention heads in each layer at the last token position (before they are projected to the residual stream dimension).

Nonlinear, neuron-specific intervention effects A key motivation for DSI is its ability to go beyond first-order effect estimates by accounting for nonlinear, neuron-specific effects and intervention effects across layers. Before proceeding to the main results, we illustrate the importance of this on the *english–french* task using Llama 3.2 (3B) by studying interventions on the most relevant neurons identified via a first-order approximation using the robustified gradient, but without iterative refinement. To this end, we randomly sample neurons and neuron pairs from the 16 neurons with the highest expected intervention effect and measure the logit of the correct token under single and pairwise interventions.

Figure 2a shows the output logit corresponding to the correct token (averaged over 100 0-shot examples from the test set) for interventions on single neurons at a time. The interventions are scaled by the average activation difference. We observe that even among the neurons with the largest expected positive effect, some reduce the output logit, and that the logit varies nonlinearly with intervention strength. Figure 2b shows the effect of intervening on pairs of neurons. We observe interaction effects, such that the maximum logit corresponds to a setting where the intervention strength differs between neurons. These observations indicate limitations of first-order effect estimation, as it does



(a) Neuron intervention effects

(b) Pairwise neuron intervention effects

Figure 2: Value of the correct output logit of Llama 3.2 (3B) for the *english-french* task, when intervening on hidden neurons. Neurons are sampled from the 16 neurons with the highest expected intervention effect, and labels correspond to their ranks. (a) shows the correct output logit as a function of single-neuron interventions, (b) as a function of interventions on pairs of neurons. Each pair consists of neurons from two different model layers.

not reliably identify neurons with a positive effect. They also point to limitations of approaches that determine a uniform scaling factor for all neurons, as even after rescaling the intervention by the average activation difference, the optimal intervention strength varies substantially across neurons, motivating neuron-level, iterative techniques such as DSI.

4.1 Evaluating DSI

For all tasks and models, we compute DSI for the neuron set sizes $n = 64$ and $n = d_{\text{model}}$ (the residual stream dimension, 2,560 to 4,096), and generate 200 0-shot prompts from the test split to evaluate the interventions. We limit ourselves to next-token prediction; if a correct output consists of multiple tokens, we evaluate whether the first predicted token is correct. We compare 0-shot task accuracy against the **baseline** model behaviour (no intervention) and **Function Vectors (FV)** (Todd et al., 2024), an approach that constructs steering vectors in the residual stream by averaging the outputs of a set of attention heads with the largest effect on task execution across multiple tasks. We select FVs for comparison because of their widespread use, and use the tasks and datasets from the original paper. Because FVs require an a priori decision regarding the intervention layer, two conditions are considered: one where the vector is added at $1/3$ of the model depth (as in Todd et al. (2024)), and another where the best-performing layer per task is used. Additionally, because the Gemma 3 model includes an extra normalisation layer after the attention block, FVs cannot be applied directly to this architecture. Therefore, we do not report FV results for this model.

Table 1: Average accuracy on the abstractive and extractive tasks in the 0-shot setting. Results for interventions on neuron set sizes 64 and d_{model} (the residual stream dimension of the model, 2,560 to 4,096). Baseline refers to model behaviour without interventions, "no rob." to no gradient robustification, "no iter." to no iterative refinement, "FV" to function vectors, "FV best" to FVs evaluated at the best-performing layer. Ranges refer to the minimum and maximum task accuracy. Best-performing methods are **bold**, second-best underlined. Across almost all conditions, DSI with gradient robustification and iterative refinement performs best.

		abstractive			extractive		
size	method	Llama 3.2	Qwen 3	Gemma 3	Llama 3.2	Qwen 3	Gemma 3
0	baseline	0.02 0.00-0.06	0.04 0.00-0.12	0.01 0.00-0.03	0.28 0.14-0.52	0.03 0.01-0.07	0.12 0.03-0.27
d_{model}	FV	0.36 0.00-0.64	0.16 0.00-0.86	/	0.43 0.22-0.96	0.41 0.04-0.97	/
d_{model}	FV best	0.57 0.00-0.92	0.33 0.00-0.97	/	0.63 0.36-0.97	0.43 0.17-0.88	/
64	no rob.	0.65 0.17-0.98	0.59 0.04-0.97	0.74 0.63-1.00	0.56 0.33-0.93	0.51 0.12-0.95	0.78 0.42-1.00
64	no iter.	0.66 0.27-0.97	0.53 0.09-0.94	0.72 0.57-1.00	0.73 0.26-0.99	0.65 0.35-1.00	0.74 0.35-1.00
64	DSI	0.89 0.74-0.98	0.87 0.70-0.98	0.86 0.70-1.00	0.98 0.95-1.00	0.98 0.97-1.00	0.96 0.94-0.99
d_{model}	no rob.	0.85 0.69-0.94	0.60 0.10-0.93	0.85 0.73-0.99	0.86 0.46-0.98	0.81 0.48-0.99	0.80 0.31-1.00
d_{model}	no iter.	0.84 0.64-0.99	0.80 0.67-0.95	0.88 0.76-0.99	0.85 0.32-0.99	0.83 0.45-1.00	0.85 0.35-1.00
d_{model}	DSI	**0.90** 0.73-0.98	**0.89** 0.67-0.98	**0.89** 0.74-1.00	**0.99** 0.98-1.00	**0.99** 0.98-1.00	**0.99** 0.98-1.00

We also compare against two additional conditions evaluating iterative refinement and gradient modifications: (1) **no iterative refinement**: the n interventions with the largest effects as approximated via

first-order Taylor expansion are used (i.e., the starting values in Algorithm 1), and one joint scaling factor $s \in [0.01, 50]$ is determined for all interventions. (2) **no gradient robustification**: the same as (1), but in addition, the first-order effects are approximated without the gradient modifications described in Section 3.1.

Main results Table 1 shows the resulting accuracy from applying DSI in the 0-shot setting across models, averaged over the two task categories (abstractive and extractive, 6 tasks each), for the intervention set sizes 64 and d_{model}. Across model families, DSI substantially improves task accuracy over the no-intervention and FV baselines, with interventions on 64 neurons achieving average accuracies ranging from 0.86 to 0.98 across settings. We observe that the accuracy on 64 neurons almost matches the full model dimension, indicating that sparse interventions can be effective. The results further indicate that both gradient robustification and iterative refinement contribute consistently to improved intervention performance.

4.2 Sparse Interventions

Figure 3 extends this analysis by showing accuracy across all 12 tasks as a function of set sizes from 1 to 256 for Qwen 3 (8B), including conditions without gradient robustification and without iterative refinement. Additional results for other models are provided in Appendix B. Shaded regions correspond to 95% Wilson confidence intervals (Wilson, 1927), taking into account uncertainty due to the limited number of test samples. Across settings, particularly for small set sizes, DSI consistently achieves the highest task accuracy, and both iterative refinement and gradient robustification are necessary for high performance, with removal of either degrading task accuracy. These results overall demonstrate that across a range of tasks, interventions on as few as 4–32 neurons activate task-solving behaviour, highlighting DSI's potential for identifying sparse and effective neuron-level interventions.

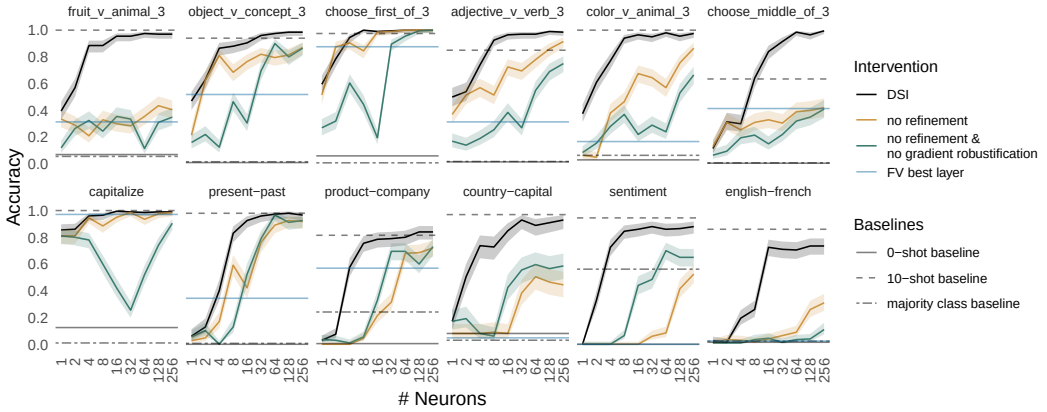


Figure 3: 0-shot test accuracy for interventions across set sizes for extractive (top row) and abstractive (bottom row) tasks. Results shown are for Qwen 3 (8B); additional results across models are provided in Appendix B. Shaded regions correspond to 95% confidence intervals. 0-shot and 10-shot baselines correspond to ICL performance without intervention; the majority-class baseline to the proportion of the most common correct outputs, and FV best layer to Function Vectors at the best-performing layer.

4.3 Distributed Interventions

To assess the relative effectiveness of distributed versus layer-localized interventions, we applied DSI restricted to a single layer at one-third of model depth, following Todd et al. (2024). Figure 4 presents the resulting accuracy differences between single-layer and distributed interventions, averaged across all extractive and abstractive tasks. For neuron set sizes of 64 or greater, effective interventions can be achieved within a single layer. For smaller set sizes, distributed interventions provide a substantial improvement in accuracy. These results suggest that the design of sparse interventions requires methods that can simultaneously estimate interventions across

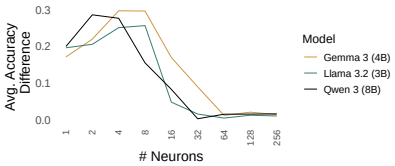


Figure 4: Average accuracy difference between single-layer and distributed DSI interventions across all 12 tasks

multiple layers. Appendix G presents further results on the distribution of identified neurons across attention heads. For instance, at set size 64, the selected neurons span a variety of attention heads across layers, primarily in middle and later layers, highlighting that neuron selection is distributed across the model and varies across tasks and models.

5 IDENTIFYING DISTRIBUTED CONCEPT REPRESENTATIONS

Open-ended concept steering Steering open-ended generation presents a challenging setting for model interventions: methods must alter specific properties of a response while preserving instruction following and fluency. To generalise our evaluations of DSI to autoregressive settings, we examine how the incorporation of concepts characterised by short textual descriptions can be achieved through distributed neural computation. We apply DSI to AxBench (Wu et al., 2025a), a standardised benchmark evaluating how well steering methods incorporate concepts during prompt answering. Each concept corresponds to a sparse autoencoder feature extracted from Gemma-2 (Gemma Team, 2024; Lieberum et al., 2024). We apply DSI to concepts extracted at layer 20 for two model sizes (2B and 9B), with 500 concepts per model size. For each concept, the dataset includes 72 positive examples generated with GPT-4o-mini instructed to include the concept and 72 negative examples from model outputs without intervention. For **evaluation**, we follow the AxBench setup, including their evaluation prompts and templates. An LLM judge[1] evaluates responses to 10 randomly sampled instructions from AlpacaEval (Li et al., 2023), using 5 of the instructions to calibrate the strength of the steering interventions, and the remaining 5 to evaluate steering performance. Each response is rated as the harmonic mean of three dimensions: (1) *Concept*: how strongly is the concept present in the response, (2) *Instruct*: how well is the prompt's instruction followed, and (3) *Fluency*: how fluent is the response. The scores on each dimension range from 0 to 2; if any of the dimensions is rated as 0, the overall steering score is set to 0. Details on adapting DSI to the generative setting can be found in Appendix H.

Effective generative steering with DSI We evaluate DSI across neuron set sizes from 64 to the full model dimension (2304/3584 for Gemma-2 2B/9B). As shown in Figure 5 (A), DSI achieves similar performance to recent dense steering methods when matching the number of neurons to the latent model dimensionality d. Limiting the neuron set size to 1024, roughly half the neuron budget for the 2B model and one third for the 9B model, maintains steering performance and demonstrates that effective steering can be achieved with a substantially reduced intervention footprint. Reducing the neuron set further reveals a trade-off between steering performance and sparsity, with DSI using 64 neurons still achieving performance on par with dense steering baselines.

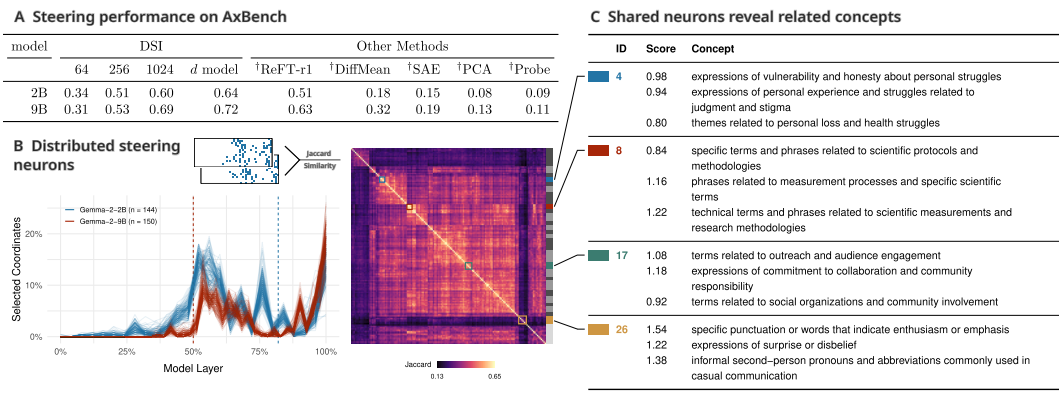


Figure 5: DSI interventions on AxBench concepts. **A** Steering performance of DSI on AxBench for different set sizes, d model (the residual stream of the model) is equal to 2304/3584 neurons for the 2B/9B parameter model. †From Wu et al. (2025a), Table 2. **B** Distribution of selected neuron coordinates across layers for DSI interventions on 256 neurons. Each line corresponds to a text concept; concepts are filtered for an AxBench score above 0.75. Dashed lines signify layer 20; the concept extraction layer. **C** Concepts are clustered by the Jaccard similarity of the intervention sets. Shown are example concepts with the highest average similarity within the cluster; "Score" corresponds to the AxBench steering score.

[1]Version `gpt-4o-mini-2024-07-18`, accessed via the OpenAI API.

Layerwise distribution of steering neurons Next, we focus on interventions of size 256, balancing sparsity and performance, on the 9B model. To study concept steering in natural language generation, we focus on the text genre and retain 150 concepts with an overall steering score above 0.75, ensuring that the corresponding interventions successfully induce the target concept. Figure 5 (B) shows the distribution of intervened-on neurons across layers for these concepts, highlighting that most steering neurons are located in late mid to late layers, with almost no interventions in the first third of the model. This pattern is remarkably consistent across concepts: we identify 81 neurons that are present in more than 80% of the interventions, despite the concepts spanning diverse categories, including grammatical and syntactic patterns, semantic descriptions, and pragmatics (see Table 6 in the Appendix). This suggests that these neurons support shared computational processes that coordinate concept expression in text generation, mirroring related findings for task switching at the level of attention heads (Todd et al., 2024).

Steering neurons capture concept relationships We next examine whether concepts with higher overlap in their intervened-upon neuron sets tend to be semantically related. We quantify the overlap between intervention sets using the Jaccard index, defined as the number of shared neurons divided by the total number of unique neurons in the two sets. After performing hierarchical agglomerative clustering, we identify 26 clusters covering 135 concepts, with 15 concepts left unassigned (see Appendix H.2 for details on the clustering procedure, a list of all identified clusters and concepts and the results on the 2B model). We observe that concepts group into diverse clusters, with most clusters exhibiting high semantic coherence. Figure 5 (C) providing examples across topics such as personal struggles, science, or informal speech. This indicates that neuron sets identified solely by their ability to induce a target concept also capture meaningful relationships across concepts, providing a link between observable model behaviour and its underlying computation.

Overall, these results demonstrate that DSI identifies sparse and distributed interventions that retain strong performance for steering open-ended text generation. The concentration of interventions in specific layers and sharing of neurons across interventions suggests that concept steering relies heavily on a relatively small set of model components. Moreover, the semantic coherence of clusters derived solely from intervention-set overlap indicates that concepts with similar intervention signatures tend to be semantically related, suggesting that the selected model neurons reflect concept-specific structure.

6 Discussion

We introduced *Distributed Sparse Interventions (DSI)*, a novel steering method for identifying neuron-level interventions. By operating directly on individual neurons rather than latent-space vectors, DSI can effectively capture interactions and nonlinear effects between neurons while providing fine-grained control over intervention strength across the model. We demonstrated that DSI enables successful steering in both abstractive and extractive task settings using surprisingly small sets of 4–32 neurons distributed across model layers, and extends effectively to concept steering in generative settings. While DSI selects intervention neurons solely based on their ability to induce the desired target behaviour, we find that greater neuron-level overlap between intervention sets corresponds to greater functional relatedness, suggesting that neuron-level sets reveal a correspondence between observable model behaviour and underlying computation. Thus, although task-relevant neurons are distributed across the model, their participation in steering appears structured rather than arbitrary.

By modelling nonlinear causal effects in steering vectors, Bigoulaeva et al. (2026) recently demonstrated that localised interventions at specific layers are involved in modulating task circuits. Our results provide further evidence for localised information processing at the neuron level. Characterising neuron-level computations may enable more precise and robust steering by targeting specific computational components rather than inducing global shifts with potentially entangled downstream effects, while also making interventions more interpretable and amenable to safeguarding.

Limitations Our evaluation covers a selected set of models and tasks, and the effectiveness of the identified intervention patterns across other architectures and steering settings requires further evaluation. For open-ended generation, future work could explore combining DSI and token-dependent intervention strengths or alternative objectives, such as preference-based losses. Finally, DSI is computationally more intensive than non-iterative steering methods.

AI USE STATEMENT

In this work, we used generative AI to implement methods, design or provide feedback on research methodology or experiments, and support qualitative and thematic data analysis. AI tools were not used to help develop theoretical models or conceptual frameworks, propose or refine hypotheses, assist with translation, and interpret results. AI tools were not used to clean and reformat datasets or generate novel synthetic data sets, but we are using datasets from prior work that are generated with the help of AI. The categories formulate mathematical claims, provide critical ingredients for proving mathematical claims, and assist in the writing of proofs are not applicable to this work.

Additionally, we used AI tools to create or edit software code, edit a research paper to improve readability, create or modify scientific figures or images, summarize or analyse existing literature, and search for information. We have reviewed all AI-assisted outputs and take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We do not anticipate any ethical concerns specific to our proposed work beyond those generally associated with model steering. Fundamental research on modifying language model behaviour could inform both alignment and potential misuse. While DSI may be employed for malicious manipulation, our goal is to advance understanding and improve safeguards for language models. We do not anticiapte any direct negative impact of our method.

REPRODUCIBILITY STATEMENT

To facilitaty reproducibility, we provide details on the used LLM implementations in Appendix C, details on the dataset source in Appendix D, details on the algorithm parameters in Appendix F, and details on the usage in autoregressive settings in Appendix H.1. Code for computing DSI interventions and the repoduction of all main results will be made openly available upon publication.

REFERENCES

Reduan Achtibat, Sayed Mohammad Vakilzadeh Hatefi, Maximilian Dreyer, Aakriti Jain, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. AttnLRP: Attention-aware layer-wise relevance propagation for transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 135–168. PMLR, 2024. URL https://proceedings.mlr.press/v235/achtibat24a.html.

Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. XAI for transformers: Better explanations through conservative propagation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 435–451. PMLR, 2022. URL https://proceedings.mlr.press/v162/ali22a.html.

Aryaman Arora, Zhengxuan Wu, Jacob Steinhardt, and Sarah Schwettmann. Language model circuits are sparse in the neuron basis. *arXiv preprint arXiv:2601.22594*, 2026.

Leila Arras, José Arjona-Medina, Michael Widrich, Grégoire Montavon, Michael Gillhofer, Klaus-Robert Müller, Sepp Hochreiter, and Wojciech Samek. *Explaining and Interpreting LSTMs*, pp. 211–238. Springer-Verlag, Berlin, Heidelberg, 2022. ISBN 978-3-030-28953-9. URL https://doi.org/10.1007/978-3-030-28954-6_11.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

David Balduzzi, Marcus Frean, Lennox Leary, J P Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: if resnets are the answer, then what is the question? In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 342–350. JMLR.org, 2017.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Daniel Beaglehole, Adityanarayanan Radhakrishnan, Enric Boix-Adsera, and Mikhail Belkin. Toward universal steering and monitoring of AI models. *Science*, 391(6787):787–792, 2026.

Irina Bigoulaeva, Jonas Rohweder, Subhabrata Dutta, and Iryna Gurevych. Patches of nonlinearity: Instruction vectors in large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12209–12262, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.559. URL https://aclanthology.org/2026.acl-long.559/.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2(5):6, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.581. URL https://aclanthology.org/2022.acl-long.581/.

Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of science*, 74(5):981–995, 2007.

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.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro Rivadeneira, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *International Conference of Machine Learning*, 2020.

Zijian Feng, Tianjiao Li, Zixiao Zhu, Hanzhang Zhou, Junlang Qian, Li Zhang, Chua Jia Jim Deryl, Mak Lee Onn, Gee Wah Ng, and Kezhi Mao. Fine-grained activation steering: Steering less, achieving more. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=guSVafqhrB.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Gemma Team. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Gemma Team. Gemma 3 technical report. 2025. URL https://goo.gle/Gemma3Report.

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.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9318–9333, 2023.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. Linearity of relation decoding in transformer language models. *arXiv preprint arXiv:2308.09124*, 2023.

Or Honovich, Uri Shaham, Samuel R. Bowman, and Omer Levy. Instruction induction: From few examples to natural language task descriptions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1935–1952, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.108. URL https://aclanthology.org/2023.acl-long.108/.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2023.

Farnoush Rezaei Jafari, Oliver Eberle, Ashkan Khakzar, and Neel Nanda. Relp: Faithful and efficient circuit discovery in language models via relevance patching. *arXiv preprint arXiv:2508.21258*, 2025.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H196sainb.

Peter Langfelder, Bin Zhang, and Steve Horvath. Defining clusters from a hierarchical cluster tree: the dynamic tree cut package for r. *Bioinformatics*, 24(5):719–720, 03 2008. ISSN 1367-4803. doi: 10.1093/bioinformatics/btm563. URL https://doi.org/10.1093/bioinformatics/btm563.

Jiaoda Li, Yifan Hou, Mrinmaya Sachan, and Ryan Cotterell. What do language models learn in context? the structured task hypothesis. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12365–12379, 2024a.

Lujun Li, Peijie Dong, Zhenheng Tang, Xiang Liu, Qiang Wang, Wenhan Luo, Wei Xue, Qifeng Liu, Xiaowen Chu, and Yike Guo. Discovering sparsity allocation for layer-wise pruning of large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=rgtrYVC9n4.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 278–300, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.19. URL https://aclanthology.org/2024.blackboxnlp-1.19/.

Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. Sparse crosscoders for cross-layer features and model diffing. https://transformer-circuits.pub/2024/crosscoders/index.html, 2024. Transformer Circuits Thread.

Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. In-context vectors: making in context learning more effective and controllable through latent space steering. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 193–209, 2019.

Neel Nanda. Attribution patching: Activation patching at industrial scale. *URL: https://www. neelnanda. io/mechanistic-interpretability/attribution-patching*, 15:19, 2023.

Neel Nanda and Joseph Bloom. Transformerlens. https://github.com/TransformerLensOrg/TransformerLens, 2022.

Bruno A Olshausen and David J Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14(4):481–487, 2004. ISSN 0959-4388. doi: https://doi.org/10.1016/j.conb.2004.07.007. URL https://www.sciencedirect.com/science/article/pii/S0959438804001035.

Bruno A Olshausen et al. Principles of image representation in visual cortex. *The visual neurosciences*, 2:1603–1615, 2003.

Abhishek Panigrahi, Nikunj Saunshi, Haoyu Zhao, and Sanjeev Arora. Task-specific skill localization in fine-tuned language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Farnoush Rezaei Jafari, Grégoire Montavon, Klaus-Robert Müller, and Oliver Eberle. MambaLRP: Explaining selective state space sequence models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 118540–118570. Curran Associates, Inc., 2024. doi: 10.52202/079017-3764. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/d6d0e41e0b1ed38c76d13c9e417a8f1f-Paper-Conference.pdf.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170/.

Lorenzo Stella. ProximalAlgorithms.jl: Proximal algorithms for nonsmooth optimization in Julia, 2025. URL https://github.com/JuliaFirstOrder/ProximalAlgorithms.jl.

Jiuding Sun, Sidharth Baskaran, Zhengxuan Wu, Michael Sklar, Christopher Potts, and Atticus Geiger. Hypersteer: Activation steering at scale with hypernetworks, 2025. URL https://arxiv.org/abs/2506.03292.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Andreas Themelis, Lorenzo Stella, and Panagiotis Patrinos. Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone linesearch algorithms. *SIAM Journal on Optimization*, 28(3):2274–2303, 2018. doi: 10.1137/16M1080240.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AwyxtyMwaG.

Victor Vargas-Barroso, Jake F Watson, Andrea Navas-Olive, Alois Schlögl, and Peter Jonas. Developmental emergence of sparse and structured synaptic connectivity in the hippocampal ca3 memory circuit. *Nature Communications*, 2026.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. Finding skill neurons in pre-trained transformer-based language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11132–11152, 2022.

Edwin B Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Axbench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=K2CckZjNy0.

Zhengxuan Wu, Qinan Yu, Aryaman Arora, Christopher D Manning, and Christopher Potts. Improved representation steering for language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=VHb883Gs1u.

Zheyang Xiong, Ziyang Cai, John Cooper, Albert Ge, Vasilis Papageorgiou, Zack Sifakis, Angeliki Giannou, Ziqian Lin, Liu Yang, Saurabh Agarwal, Grigorios Chrysos, Samet Oymak, Kangwook Lee, and Dimitris Papailiopoulos. Everything everywhere all at once: LLMs can in-context learn multiple tasks in superposition. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=5hZCK4Wbex.

Fangcong Yin, Xi Ye, and Greg Durrett. Lofit: Localized fine-tuning on LLM representations. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 9474–9506. Curran Associates, Inc., 2024. doi: 10.52202/079017-0301. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/122ea6470232ee5e79a2649243348005-Paper-Conference.pdf.

A INTERACTIONS BETWEEN INTERVENTIONS

Several components in transformer models can induce interactions between interventions. For example, products (the self-attention mechanism) induce interactions between interventions on the multiplied neurons, while nonlinearities (activation functions) induce interactions between interventions on neurons connected across layers. Here, we study an example of how product terms induce such interactions, and how an iterative computation of interventions can better account for them.

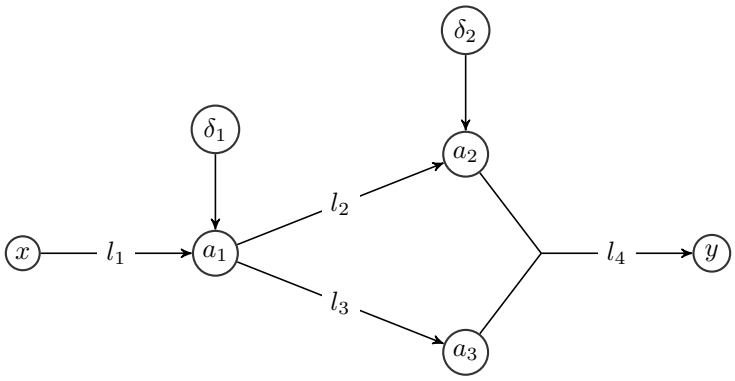


Figure 6: Example system with causal interventions δ_1 and δ_2.

Consider the model $f : \mathbb{R} \to \mathbb{R}$, $x \mapsto l_4(l_2(l_1(x)), l_3(l_1(x)))$ depicted in Figure 6. Setting $l_1 = l_2 = l_3 = \mathrm{id}$ and $l_4(a_2, a_3) = a_2 a_3$, it reduces to $f(x) = x^2$. The effects of δ_1 and δ_2 in isolation are

$$\mathcal{E}(\delta_1) = f(x|a_1 \leftarrow a_1 + \delta_1) - f(x) = (x + \delta_1)^2 - x^2 = 2x\delta_1 + \delta_1^2 \tag{4}$$
$$\mathcal{E}(\delta_2) = f(x|a_2 \leftarrow a_2 + \delta_2) - f(x) = (a_2 + \delta_2)a_3 - a_2 a_3 = x\delta_2 \tag{5}$$

Now, consider adding both effects simultaneously:

$$\mathcal{E}(\delta_1, \delta_2) = f(x|a_1 \leftarrow a_1 + \delta_1, a_2 \leftarrow a_2 + \delta_2) - f(x) \tag{6}$$
$$= 2x\delta_1 + \delta_1^2 + x\delta_2 + \delta_1\delta_2 \tag{7}$$
$$= \mathcal{E}(\delta_1) + \mathcal{E}(\delta_2) + \delta_1\delta_2. \tag{8}$$

The effect of both interventions combined is not equal to the sum of the intervention effects in isolation, since an interaction term $\delta_1\delta_2$ occurs. This interaction term is therefore also absent from the first-order Taylor approximation of the intervention effects:

$$\bar{\mathcal{E}}(\delta_1) = \partial_{a_1} f(x)\delta_1 = 2a_1\delta_1 = 2x\delta_1 \tag{9}$$
$$\bar{\mathcal{E}}(\delta_2) = \partial_{a_2} f(x)\delta_2 = a_3\delta_2 = x\delta_2 \tag{10}$$

However, this effect can be mitigated through an iterative approach. Consider the setting where the effects have been approximated and the intervention with the highest effect (e.g., δ_2) has been added to the model. Now, to add more interventions, we re-approximate at the new root point, that is, we approximate $f_{\delta_2}(x) = f(x|a_2 \leftarrow a_2 + \delta_2)$:

$$\bar{\mathcal{E}}(\delta_1) = \partial_{a_1} f_{\delta_2}(x)\delta_1 = (2a_1 + \delta_2)\delta_1 = 2x\delta_1 + \delta_1\delta_2 \tag{11}$$

and we can see that at the new root point, the approximation becomes more precise as it includes the interaction effect between the interventions.

B RESULTS FOR LLAMA 3.2 AND GEMMA 3

Figures 7 and 8 show the accuracy by intervention set size for the LLama 3.2 (3B) and Gemma 3 (4B) models, respectively. These figures are analogous to Figure 3 in the main text.

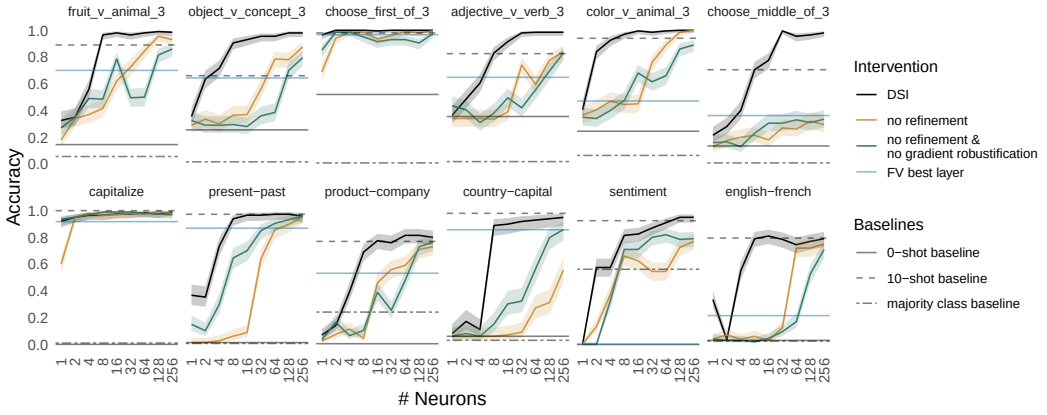


Figure 7: 0-shot test set accuracy for interventions across set sizes for extractive (top row) and abstractive (bottom row) tasks. Results shown are for Llama 3.2 (3B), shaded regions correspond to 95 % confidence intervals. 0-shot and 10-shot baselines correspond to ICL performance without intervention, majority class baseline to the proportion of the most common correct output token, and FV best layer to the function vector intervention added at the best performing layer.

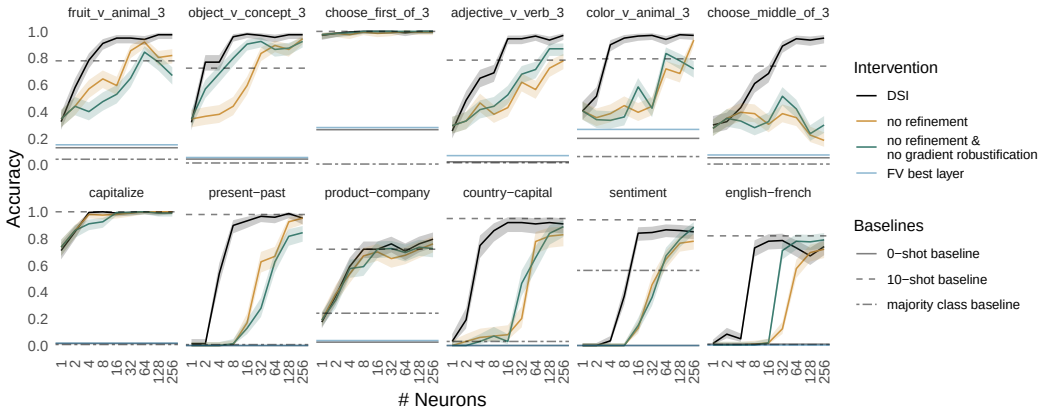


Figure 8: 0-shot test set accuracy for interventions across set sizes for extractive (top row) and abstractive (bottom row) tasks. Results shown are for Gemma 3 (4B), shaded regions correspond to 95 % confidence intervals. 0-shot and 10-shot baselines correspond to ICL performance without intervention, majority class baseline to the proportion of the most common correct output token, and FV best layer to the function vector intervention added at the best performing layer.

C MODEL DETAILS

For all models, we use the model implementations provided by TransformerLens (Nanda & Bloom, 2022). TransformerLens translates the model implementation into a standardised set of layers and centers the unembedding. Table 2 provides the corresponding TransformerLens identifiers as well as the model dimensions.

Table 2: Properties of the studied models. The total number of considered neurons for the DSI intervention is the product of model layers, heads, and the number of neurons per head.

Model	TransformerLens ID	# Layers	# Heads	# Neurons p.h.	# Neurons total
Qwen 3 (8B)	`Qwen/Qwen3-8B`	36	32	128	147 456
Gemma 3 (4B)	`google/gemma-3-4b-it`	34	8	256	69 632
Llama 3.2 (3B)	`meta-llama/Llama-3.2-3B-Instruct`	28	24	128	86 016

D TASKS DETAILS

Table 3 provides an overview of the tasks used in Section 4. We use all tasks as provided in the supplementary material to Todd et al. (2024), accessible at the accompanying GitHub repository[2] under MIT License. We split all tasks 50/50 into a training and a test set. To generate k-shot ICL examples, we follow the default template of Todd et al. (2024) and present k input-output examples follow by an example to complete as "Q: input$_1$\nA: output$_1$\n\nQ: input$_2$\nA: output$_2$\n\n[...]Q: input$_k$\nA: output$_k$\n\nQ: input\nA:". 0-shot examples are of the form "Q: input\nA:". From the training set, we generate 100 0-shot and 100 10-shot prompts to compute interventions. From the test set, we generate 200 0-shot prompts to compute the reported accuracies. All tasks contain at least 200 test set samples, except for **present-past** (146 training set and 147 test set examples) and **country-capital** (98 training and 99 test set examples) - these tasks have been trained and evaluated on fewer examples, respectively, and their confidence intervals reflect the higher uncertainty level.

Table 3: Overview of the 12 tasks used in the main experiments in Section 4.

Task	**Description**	**Example**
`capitalize`	Capitalize the first letter of a word.	*curtain* → *Curtain*
`country-capital`	Given a country, output its capital city.	*Ecuador* → *Quito*
`english-french`	Translate an English word into French.	*loved* → *aimé*
`present-past`	Convert a verb from present to past tense.	*teach* → *taught*
`sentiment`	Classify the sentiment of a short text as positive or negative.	*"Beautifully produced."* → *positive*
`product-company`	Given a product, identify the company that created it.	*Office Open XML* → *Microsoft*
`fruit_v_animal_3`	Given two animals and a fruit, identify the fruit.	*date, penguin, reindeer* → *date*
`color_v_animal_3`	Given two animals and a color, identify the color.	*eel, olive, turtle* → *olive*
`object_v_concept_3`	Given two abstract concepts and one physical object, identify the object.	*about, think, donut* → *donut*
`choose_first_of_3`	Given three words, output the first one.	*zealous, kindhearted, plant* → *zealous*
`adjective_v_verb_3`	Given two verbs and one adjective, identify the adjective.	*search, thirsty, manipulate* → *thirsty*
`choose_middle_of_3`	Given three words, output the middle one.	*quokka, hot, sail* → *hot*

Todd et al. (2024) adapted some of the tasks from earlier work: the **english-french** translation task is an adaptation from Lample et al. (2018), the **product-company** task is adapted from Hernandez et al.

[2] `https://github.com/ericwtodd/function_vectors` (commit fb9eac7b6dc707ea1475a717379916007fe448d5)

(2023), and the sentiment analysis dataset is an adaptation from Socher et al. (2013) and Honovich et al. (2023).

For Section 3.2, we generate a dataset for a **copy** task by using the inputs of the **english-french** task, and set the output equal to the input (i.e., the datapoint *loved* $\rightarrow$ *aimé* becomes *loved* $\rightarrow$ *loved*).

E LRP-MODIFIED GRADIENT COMPUTATION

To implement the modified gradients, we adapt the respective PyTorch modules implemented in TransformerLens (Nanda & Bloom, 2022). The modifications `detach` parts of the computation graph, leaving the forward pass unchanged, but alter the backward pass behaviour. Table 4 contains an overview of the applied rules and how they are implemented.

Table 4: LRP rules applied to transformer modules. $\mathrm{sg}(\cdot)$ denotes the stop-gradient (detach) operator.

Module	**Rule**	**Modified Forward Pass**
Activation function $g(a)$	Identity rule	$\tilde{a} = a \cdot \mathrm{sg}\left(\frac{g(a)}{a}\right)$
Gated MLP with activations a	Half rule	$\tilde{a} = \frac{1}{2}\, a + \frac{1}{2}\, \mathrm{sg}(a)$
Attention with α/V matrices M	Half rule	$\tilde{M} = \frac{1}{2}\, M + \frac{1}{2}\, \mathrm{sg}(M)$
RMSNormPre with activation vector $\mathbf{a}$	LayerNorm rule	$\tilde{\mathbf{a}} = \frac{\mathbf{a}}{\mathrm{sg}(\sigma)}, \quad \sigma = \sqrt{\frac{1}{d}\sum_i \mathbf{a}_i^2 + \varepsilon}$

For nonlinear activation functions g that act on individual (scalar) activations at a time (e.g., SiLU), we employ the **Identity rule** (Rezaei Jafari et al., 2024) which treats g as constant, preserving the magnitude of the effect estimate. Gated MLPs and the attention mechanism contain multiplications of activations. In these settings, we employ the **Half rule** (Arras et al., 2022; Achtibat et al., 2024) which prevents the effect estimate from doubling during the backward pass. For layer normalization implemented by RMSNormPre modules, we use the **LayerNorm rule**, as described in (Ali et al., 2022), to avoid an effect estimation collapse.

F ALGORITHM DETAILS

For the iterative refinement of the interventions, we use ZeroFPR (Themelis et al., 2018), a numerical optimisation algorithm suited for the sum of a non-convex differentiable function and a nonsmooth penalty term. To ensure sparsity of the identified neuron sets, we use the L_0 pseudo-norm ball as a penalty.

$$\iota_{B_0(n)}(s) \coloneqq \begin{cases} 0 & \text{for } \mathrm{nnz}(s) <= n \\ \infty & \text{for } \mathrm{nnz}(s) > n \end{cases}$$

Here, $\mathrm{nnz}(s)$ corresponds to the number of non-zero elements in s and n is the desired neuron set size. In our implementation of the algorithm, we follow the implementation in ProximalAlgorithms.jl (Stella, 2025). Specifically, we use a convergence tolerance of 10^{-8} with a maximum number of 50 steps, adaptive step size estimation, a maximum number of 20 line search backtracks, and L-BFGS (Liu & Nocedal, 1989) to compute the line-search directions with a memory of 5 steps. To facilitate escaping local minima near the starting values, we run the algorithm 10 times with perturbed starting values. For the first run, we start with $s_0^1 = \mathbf{1}(\mathrm{topn}(\mathbf{e}))$, i.e., we set all parameters to 0 except for the top n largest effects approximated with a first-order approximation (as described in Section 3.2). For successive runs, we start with random noise $s_0^i \sim \mathcal{N}(0, 0.3)$. We then take an initial step $\mathrm{topn}(s_0^i + 1.5\ \gamma \nabla f(s_0^i))$ where γ is the initial step size guess of ZeroFPR. This allows to sample different gradients at points close to s_0^1, while a factor of 1.5 applied to the initial step size encourages the algorithm to explore regions beyond the limits of the algorithm's usual first step. Successive steps are taken without step size rescaling.

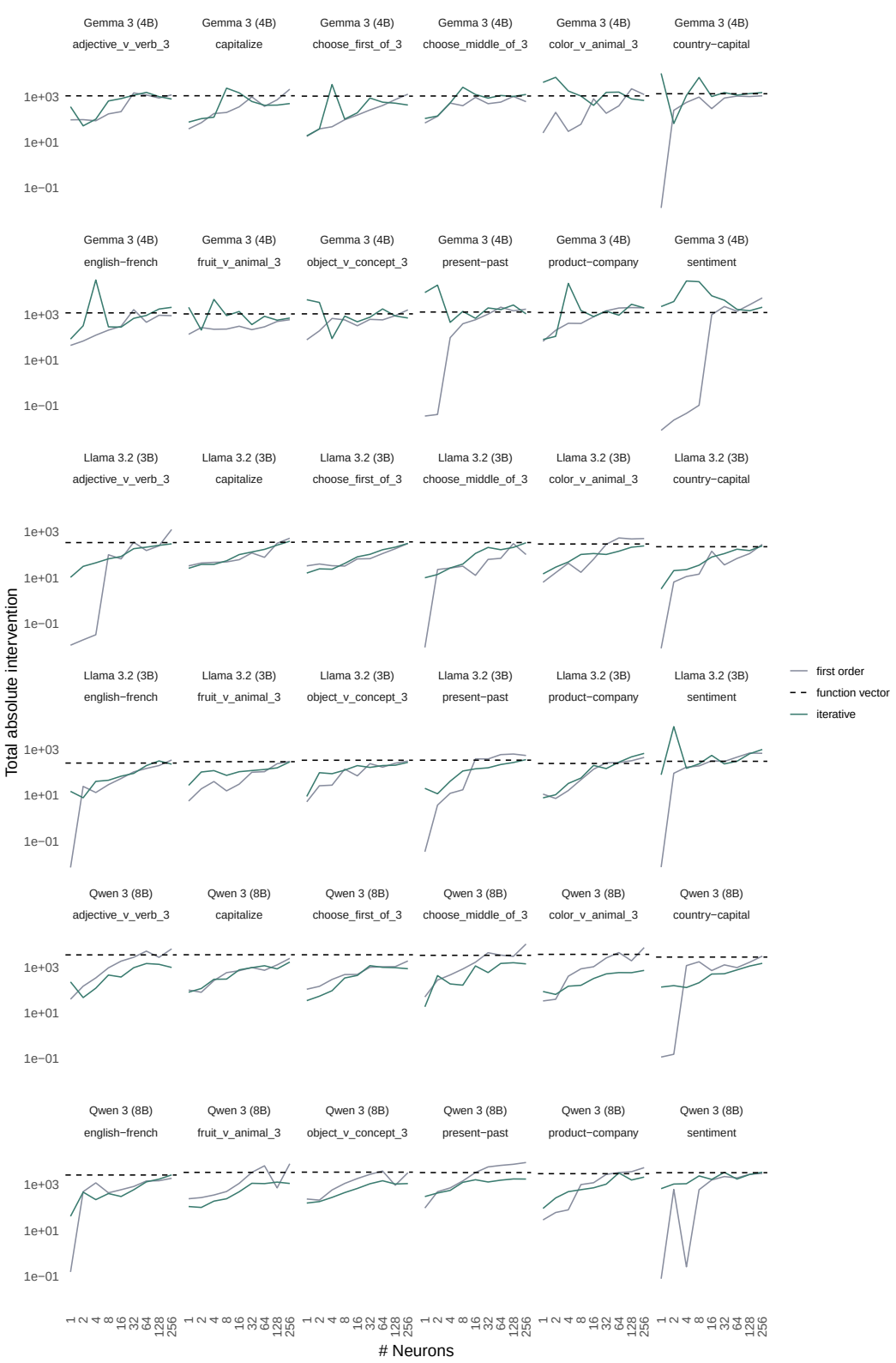


Figure 9: Absolute intervention strength for the first-order, iterative, and function vector interventions, summed over neurons.

Intervention comparison We briefly analyze the difference in interventions identified by the first-order approximation and the iterative algorithm. Figure 10 shows the overlap of the neuron intervention sets of the two intervention methods across tasks and models. While there is overlap, a substantive fraction of the neurons chosen by the iterative procedure are not among the top neurons as identified by the first-order approximation, indicating that the iterative search indeed significantly changes the intervention set.

Additionally, to verify that differences in task performance between first-order and iterative interventions do not stem from widely different total intervention strengths, rather than differences in the selected neurons, we show in Figure 9 the total intervention strength for the first-order, iterative, and function vector interventions. While there are differences in intervention strength, there is no general trend (such as iterative or first-order interventions always being stronger).

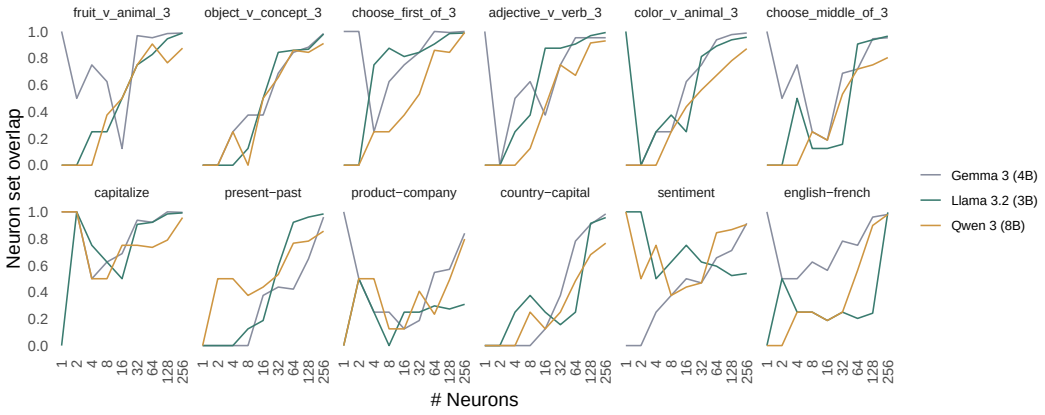


Figure 10: Overlap of neuron sets identified by first-order approximation and the iterative algorithm.

G INTERVENTION DISTRIBUTION

In this section, we investigate the distributed nature of DSI interventions, i.e., whether the identified neuron sets distribute widely across layers and attention heads, or whether they are rather tightly localised within the models. Figure 11 shows that DSI interventions, over all tasks, are distributed across many model components (i.e., attention heads), with even very small interventions being spread across multiple heads. Figures 12, 13, and 14 show the involved layers and heads for DSI interventions of a set size of 64 for each individual task. The interventions are distributed across layers, with a preference for early-middle to late layers in the Qwen 3 (8B) and Gemma 3 (4B) models.

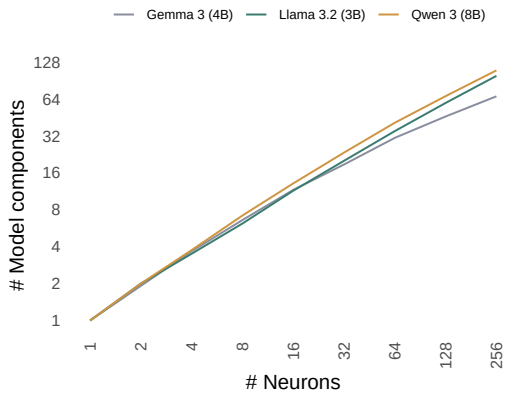


Figure 11: Number of intervened-on model components (attention heads), identified by DSI, per neuron set size, averaged across tasks.

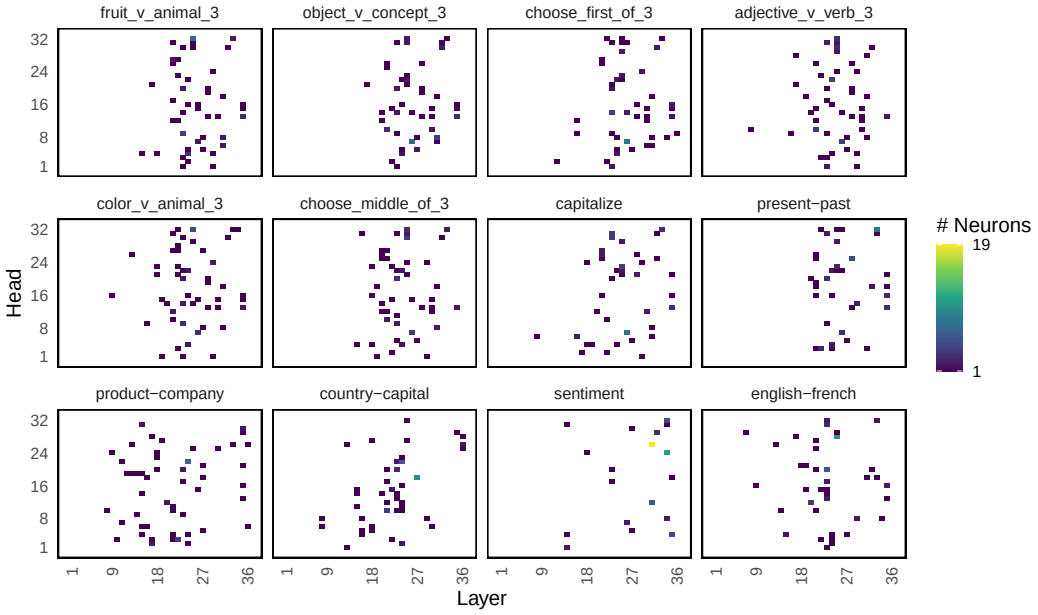


Figure 12: Number of intervened-on neurons, identified by DSI, within layers and heads for Qwen 3 (8B) and a set size of 64.

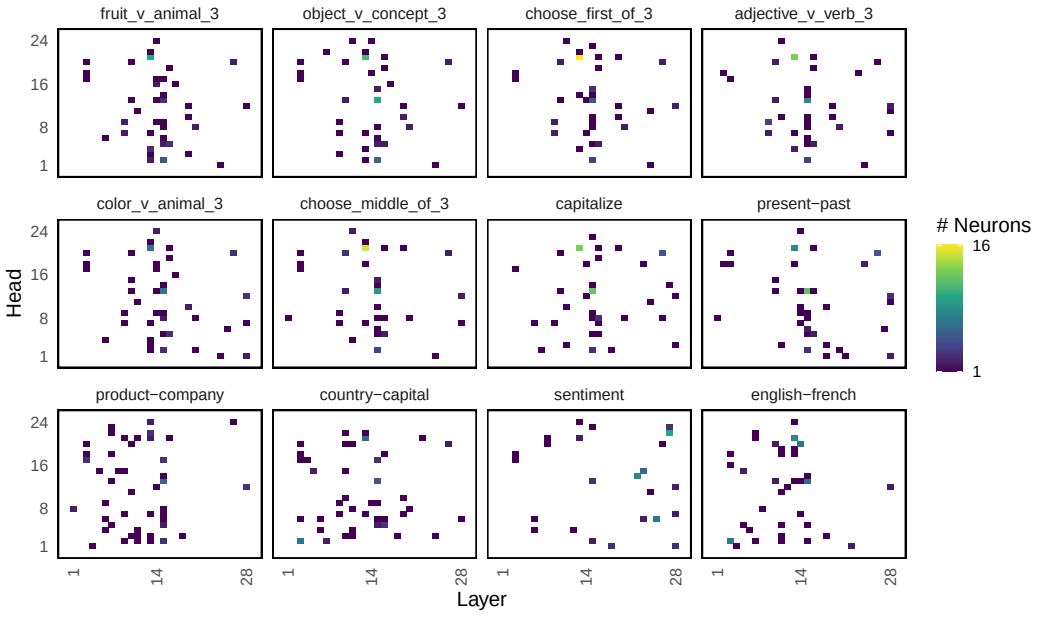


Figure 13: Number of intervened-on neurons, identified by DSI, within layers and heads for Llama 3.2 (3B) and a set size of 64.

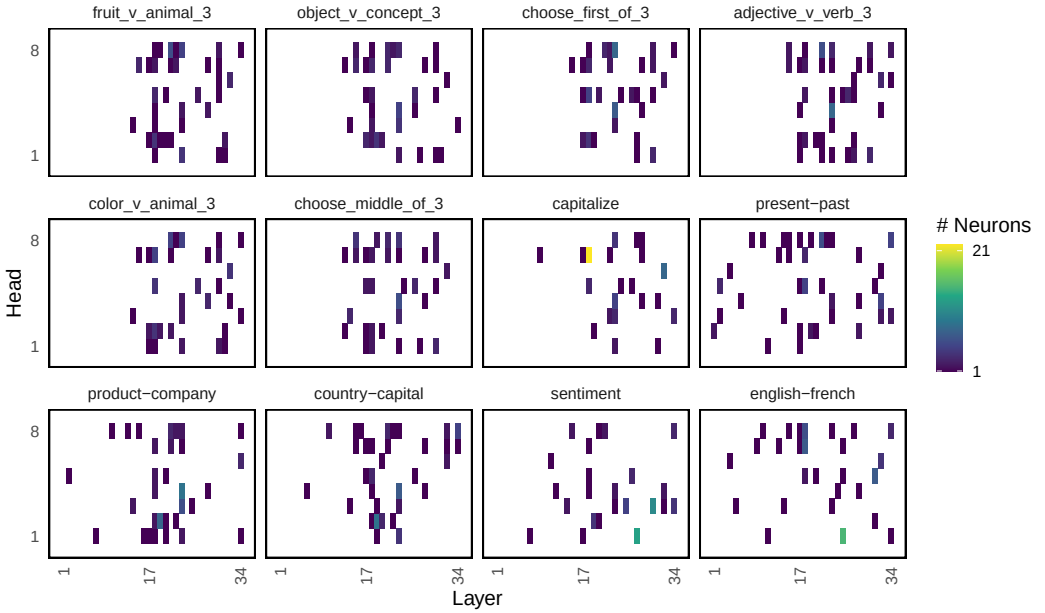


Figure 14: Number of intervened-on neurons, identified by DSI, within layers and heads for Gemma 3 (4B) and a set size of 64.

H AxBench application details

H.1 Adapting DSI to the AxBench setting

To compute DSI interventions on AxBench, we again consider the outputs of all attention heads in each layer. There, we compute the average activation difference between the positive and negative examples, and as the objective for the iterative refinement, we compute the cross-entropy with the positive examples. To save compute, we reduce the number of random restarts to 2 and the number of ZeroFPR iterations to 30. Since the output consists of multiple tokens now, we average over the output token span to compute activation differences. For the cross-entropy objective, we use the positive examples as inputs, averaging the cross-entropy of the next token predictions conditional on the prior example tokens (i.e., teacher forcing) over all response-region tokens of an example. This value is then averaged over the concepts' positive examples. For steering, we apply the interventions at each token position except for the very first `<bos>` token.

H.2 Clustering details

We perform hierarchical agglomerative clustering with average linkage, using the R function `hclust`, with 1 - Jaccard index as the distance matrix. To identify clusters from the resulting dendrogram, we use dynamic tree cutting (Langfelder et al., 2008) as implemented in the R package `dynamicTreeCut`. Since the number of clustered data points is comparatively low, we set the minimum cluster size and the `deepSplit` parameter to 3, favouring many small clusters, and do not merge unassigned concepts into the nearest cluster. Tables 5 and 6 contain a full list of identified clusters and their concepts.

Table 5: Intervention clusters of Gemma-2-2B for an intervention size of 256; - = not assigned to any cluster.

Cluster	Score	Concept
1	1.38	references to personal pronouns and possessive adjectives
	1.32	expressions of enjoyment and fun
	1.22	greetings and informal salutations
	1.20	statements that express personal beliefs or experiences
	1.08	personal pronouns and their usage in relation to self-reference

Cluster	Score	Concept
	1.04	sentences that emphasize personal reflections or opinions
2	1.18	statements, commands, or instructions related to user interactions and comments
	0.84	contractions indicating negation
	0.78	calls to action for clicking links or buttons
3	1.12	phrases related to navigating directions and movement
	0.98	words related to signals and identifiers in various contexts
	0.78	mentions and variations of the word "distance."
4	1.00	references to folder revisions and their management
	0.92	diagrams and schematic representations related to workflows, communication systems, and protocols
	0.92	terms related to electrical charge and currents
	0.84	quantum-related concepts or terms
	0.84	terms and phrases related to servers and server errors
5	1.22	references to Google and its associated services
	1.08	information about events and activities in a community, including announcements, tours, and concerts.
	0.80	references to websites and web-related services
6	0.88	connections and associations in text regarding media influence or misinformation
	0.84	language related to controversies and criticism in political contexts
	0.78	phrases related to criticism and disapproval towards actions or policies
7	1.22	references, citations, and numerical data in a text
	1.12	detailed references and citations in academic writing
	1.00	references to authors or contributors in a scientific context
8	1.12	terms related to news and its various formats
	0.98	elements related to document structure and formatting
	0.84	references to sources and reporting in news
	0.80	references to figures within a document
9	1.14	references to legal documents or official communications
	1.08	concepts related to legal obligations and consequences
	0.98	terms related to trading and financial instruments
	0.94	expressions related to legal or financial obligations
	0.92	references to legal procedures and rights
	0.90	references to legal terms and actions
	0.90	terms related to copyrights, licenses, and permissions in software documentation.
	0.88	references to temperature measurements and conditions in scientific contexts
	0.84	thematic elements related to legal or contractual obligations
	0.78	financial terms and concepts related to loans and mortgages
10	1.32	mentions of significant market factors or trends
	1.12	terms related to revenue and financial regulations
	0.92	titles and keywords related to economic analysis and market dynamics
11	0.98	terms related to trade negotiations and governmental decisions
	0.92	terms related to governance, citizenship, and roles in societal structures
	0.84	titles and roles within academic or professional contexts
12	1.08	references to brands, products, or specific entities in various contexts
	1.04	phrases related to scientific advancements and their implications
	1.04	references to media, technology, and medical terminology
	0.88	specific patterns related to digital communication and media formats
13	1.22	phrases related to problem-solving and decision-making challenges
	1.10	concepts related to decision-making and management outcomes

Cluster	Score	Concept
	1.04	phrases relating to performance expectations and potential improvements in a competitive context
	0.80	patterns related to warnings or alerts in a procedural context
14	1.34	terms related to procedures and techniques
	1.32	scientific terms related to research findings and their implications
	1.14	scientific concepts and terminology related to uncertainty and measurement
	0.88	notes or signals relating to metrics, stats, and data entries
	0.84	elements related to scientific data and results
15	1.18	references to advocacy and support services
	1.16	terms related to environmental management and pollution reduction
	1.04	terms related to environmental policies and factors
	0.92	references to health-related professionals and various stakeholders involved in healthcare discussions
	0.88	phrases related to health communication and discussion
	0.88	discussions related to educational preparation and career readiness
	0.80	key concepts and guidelines related to risk communication in public health research
16	0.96	references to health-related terms, particularly in the context of potential medical consequences
	0.92	references to species or medical terms related to cancer treatments or conditions
	0.88	instances of sports-related injuries
17	0.94	references to time frames and chronological events
	0.90	references to political events and figures
	0.88	references to significant political events or figures
	0.80	historical places or contexts that are significant to specific events or landmarks
18	1.38	elements related to narrative and storytelling
	1.20	elements related to immersive reading experiences and emotional engagement with characters
	1.04	references to different worlds, realities, or fantastical settings within narratives
	0.92	references to influential artists and creative inspirations in the context of writing and music
19	1.32	references to family relationships and dynamics
	1.20	references to authority figures or personal experiences related to persuasive narratives
	0.92	references to children and pediatric contexts
	0.92	references to individuals and their experiences or identities
20	1.38	terms related to personal relationships and social dynamics
	1.14	references to relationships and social dynamics
	1.02	phrases related to improving relationships and business collaboration
21	1.04	phrases that describe essential components or elements related to care, sustenance, and values
	0.98	phrases related to significant achievements or impactful moments
	0.92	words and phrases associated with winning and competition
	0.88	phrases related to possession or experiences over time
22	1.14	concepts related to hidden or concealed elements
	1.08	phrases related to awareness and realization
	1.08	references to navigation and location tracking
	1.04	instances of experiences that are novel or unfamiliar
	0.92	references to trips or journeys
23	1.08	negative language and themes related to distrust and conflict

Cluster	Score	Concept
	1.02	concepts related to struggle and self-control, particularly regarding desires and instincts
	0.94	key terms and phrases related to turmoil, crises, and significant conflicts in various contexts
	0.88	phrases conveying medical conditions and the presence of pain
24	1.60	phrases that express comparisons or similarities
	1.04	names or references to specific individuals or characters
	1.02	proper nouns, particularly names and locations
	0.78	names of people or entities associated with specific contexts or citations
25	1.38	phrases that convey necessity or importance
	1.28	phrases indicating conditional or preceding circumstances
	1.22	conditional phrases indicating necessity or circumstances
	1.18	phrases indicating belief, assertion, or opinion about subjects
	1.14	conditional phrases starting with "if."
26	1.32	phrases expressing uncertainty or opinion
	1.12	phrases that express skepticism or critical viewpoints
	0.98	statements about opinions, assertions, and disclaimers regarding information and its accuracy
27	1.12	instances of the word "next."
	1.08	phrases that indicate the beginning of actions or events
	0.96	phrases related to repetitive actions or follow-up events
	0.90	actions related to movement or directives, particularly in tense contexts
28	1.22	terms related to maintaining or preserving a state or condition
	1.16	qualifiers and intensifiers that modify the degree of an adjective or adverb
	1.08	statements of existence or presence
	1.02	various types of items or features, often describing characteristics or attributes
	1.00	the presence of the word "public."
	0.94	terms related to differentiation, separation, and distinction
	0.92	verbs that indicate ease or improvement in processes or experiences
	0.88	terms related to adaptability and adjusting to changing conditions
	0.84	words and phrases related to service offerings and product availability
29	1.44	instances of comparisons and contrasts in statements
	1.16	conjunctions and linking words that facilitate the connection between ideas
	1.12	words and phrases indicating comparison or qualification
	1.02	the word "such" in various contexts, often indicating conditions or examples
	1.02	phrases indicating comparisons or relationships between two entities or elements
	0.92	the word "much" in various forms and contexts
–	1.44	questions beginning with "how."
	1.28	phrases related to providing information or guidance about procedures
	1.22	references to candy and confectionery products
	1.14	references to menus and related menu items within content
	1.10	technical or operational terms related to project management and documentation
	0.98	references to farms and agricultural concepts
	0.98	references to standardized testing
	0.96	punctuation marks, particularly semicolons and parentheses
	0.94	contextual details concerning costs and pricing
	0.92	the presence of single quotation marks
	0.92	descriptive elements related to architecture and physical spaces
	0.92	numerical citations or references within the text
	0.84	specific formatting or markers in the text structure
	0.84	elements related to procedural guidelines and recommendations

Cluster	Score	Concept
	0.84	factors related to communication system performance and scientific research needs
	0.80	references to specific teams and player statistics in sports

Table 6: Intervention clusters of Gemma-2-9B for an intervention size of 256; - = not assigned to any cluster.

Cluster	Score	Concept
1	1.32	elements related to validation and classification criteria in a structured data context
	1.22	terms related to accumulation and buildup of substances
	1.20	sentence-ending punctuation and transitional phrases
	1.12	references to security-related terms
	1.12	references to real estate and property ownership
	1.08	past tense verbs indicating actions or events
	1.04	references to educational background and institutions
	0.96	references to geographical locations, specifically cities
	0.92	references to concealed or enclosed spaces and protective casings
	0.92	terms related to winning or achieving victories in competitions or games
	0.84	references to relationships and joint decisions
	0.84	components related to educational sessions and presentations
	0.80	references to big data technologies and hadron physics
	0.78	technical specifications and requirements related to software and hardware
2	1.64	words that indicate personal pronouns or first-person references
	1.38	conversational phrases that indicate problem-solving or assistance
	1.14	instances of direct speech or quotation
	1.02	phrases indicating instructions or tutorials involving the word "to."
	0.98	the presence of personal pronouns and certain numerical values in the text
3	1.26	references to divine beings and celestial objects
	1.10	references to nighttime or nocturnal themes
	0.84	words and phrases related to elegance and elaborate style
4	0.98	expressions of vulnerability and honesty about personal struggles
	0.94	expressions of personal experience and struggles related to judgment and stigma
	0.80	themes related to personal loss and health struggles
	0.78	references to difficult or distressing situations and the associated emotional responses
5	1.32	elements of personal experience and emotional state
	1.18	pronouns and references to the self or people in a narrative context
	1.08	phrases related to past events and personal connections
6	1.48	terms related to personal interests and emotional experiences
	0.94	positive sentiments and descriptions of enjoyable experiences
	0.84	emotional expressions and responses in relation to movies
	0.84	phrases that contain references to the concept of art or artistic content
	0.84	various expressions of emotional or sensory perceptions
7	1.32	references to musical instruments or compositions
	1.22	phrases and concepts related to gastronomy and culinary experiences
	1.16	references to food items and catering options
	1.12	phrases related to visiting or engaging with specific activities or locations
	1.08	references to social gatherings and events
	1.04	references to color and backgrounds in design contexts
	0.92	terms related to online gambling and casinos

Cluster	Score	Concept
	0.84	references to music and songs
	0.84	references to dressing or attire
8	1.22	technical terms and phrases related to scientific measurements and research methodologies
	1.16	phrases related to measurement processes and specific scientific terms
	0.84	specific terms and phrases related to scientific protocols and methodologies
	0.78	terms related to academic research, databases, and technical fields
9	1.02	specific terms related to efficiency and regulation within systems
	0.94	key structural elements and functions within technical or scientific contexts
	0.88	specific terms related to technical aspects and functionalities in various contexts
10	1.30	keywords related to scientific testing and chemical analysis
	0.94	technical vocabulary related to semiconductor components and processes
	0.92	technical terminology and data related to particle physics experiments
	0.88	terms and concepts related to biology and medicine
	0.78	terms related to biochemical compounds and their effects
	0.78	terms related to neuroscience and brain function
11	1.18	terms related to energy transfer and conversion processes
	0.88	references to "Digital Twins" and related concepts
	0.84	terms and concepts associated with cloud computing and virtualization services
12	1.38	words or phrases related to achievement and effectiveness
	1.28	intensifiers that emphasize magnitude or degree
	1.26	words associated with importance or significance
	0.98	expressions related to admiration and respect for individuals
13	1.10	phrases that express awareness or knowledge about a situation
	1.08	references to the user or their actions
	1.04	sentences reflecting understanding and agreement in discussions
14	1.32	phrases indicating future events or actions
	0.96	instances of the word "provided" and its variations
	0.94	phrases indicating obligation or necessity
15	1.22	elements that indicate comparisons or contrasts between subjects
	1.22	phrases indicating causation and the effects of specific actions or conditions
	1.22	instances of the word "including."
	1.14	different forms of combinations or integrations between multiple elements or factors
	1.08	phrases that indicate additional information or context being included
	1.02	phrases indicating relationships or associations within a context
	0.96	aspects of clarification and detail in communication
	0.94	terms indicating the quality or effectiveness of performance and health outcomes
	0.84	information related to categories or classifications
16	1.22	instances of the word "show" and its variations
	1.16	phrases indicating frequency and patterns of behavior
	1.08	prepositions and phrases of location and context
	1.08	the term "wide" in various contexts
	0.92	references to physical contact or proximity between objects or people
	0.92	repetitive sequences or patterns in data
	0.92	instances of the word "face" and its variations, indicating confrontations or challenges faced by subjects in the text
	0.84	phrases related to totality or absence
	0.80	instances of the term "success"
	0.80	occurrences of the word "on"

Cluster	Score	Concept
17	1.42	phrases that emphasize inclusion and represent a sense of belonging
	1.18	expressions of commitment to collaboration and community responsibility
	1.12	concepts and discussions related to creativity and innovation
	1.08	terms related to outreach and audience engagement
	0.92	terms related to social organizations and community involvement
18	1.12	references to churches and religious institutions
	0.96	references to significant events or actions related to time and people
	0.94	references to different religious affiliations and conditions related to mental health
	0.94	terms related to voting and election processes
	0.88	references to power dynamics and political figures
	0.84	terms related to mental health issues and emotional distress
19	1.18	references to templates and user-generated content creation processes
	1.00	links and references to accounts and privacy-related content
	0.88	terms related to Creative Commons licensing
20	1.28	references to businesses and industries
	1.08	references to specific individuals or entities involved in reported events
	0.98	references to people or groups in a social context
	0.96	mentions of institutional or organizational affiliations
	0.92	references to government departments or agencies
	0.90	references to specific geographic locations and areas in the context of economic discussion
	0.84	mentions of government officials and institutional affiliations
21	1.12	references to reports, surveys, or studies that include data and conclusions
	1.04	references to academic journals and publications
	0.98	references to data sets and comparative analysis in research contexts
	0.94	references to studies or data related to medical conditions and their treatments
	0.80	data and findings related to research outcomes and analyses
22	1.24	terms and concepts related to specific fields, particularly in law, health, and technology
	1.04	legal terminology and references to specific cases or patients
	0.98	keywords related to legal and political concepts
	0.84	phrases related to political and legal context
	0.84	references to legal proceedings and court orders
23	1.32	terms and phrases related to transaction processes and review mechanisms
	1.18	references to systems and their components
	1.12	terms related to eligibility and selection processes
24	1.08	phrases indicating a need for improvement or development in technology
	0.92	references to education and training backgrounds
	0.88	terminology related to restrictions and health measures
	0.88	references to changes in rules and policies affecting guidelines or regulations
25	1.26	the word "either" in various contexts
	1.18	questions and statements challenging beliefs or assumptions
	0.98	various types of conditional and contrasting phrases
26	1.54	specific punctuation or words that indicate enthusiasm or emphasis
	1.50	imperative phrases and calls to action
	1.38	informal second-person pronouns and abbreviations commonly used in casual communication
	1.32	playful and onomatopoeic sounds or expressions
	1.22	expressions of surprise or disbelief
	0.98	conversational expressions or discourse markers

Cluster	Score	Concept
–	1.38	inquiries and questions about objects, actions, or concepts in various contexts
	1.32	references to internet links or URLs
	1.28	elements related to data visualizations and plots
	1.12	terms related to food and energy prices
	1.08	terms related to defects and corrective actions in reports
	1.08	occurrences of significant transitional phrases and markers
	1.02	phrases related to concerns and the consequences of inaction
	0.98	references to specific items or products related to storage solutions and organization
	0.98	phrases related to formal requests and demands
	0.96	recommendations for actions, particularly regarding media and activities
	0.90	terms related to research methodologies and educational frameworks
	0.80	proper nouns and specific names related to characters, objects, or places
	0.80	formal language related to methodology and research findings
	0.78	elements within structured lists or updates
	0.78	phrases related to service and customer support