Neurons Speak in Ranges: Breaking Free from Discrete Neuronal Attribution

Muhammad Umair Haider*
Department of Computer Science
University of Kentucky
Kentucky, USA
muhammadumairhaider@uky.edu

Hammad Rizwan†
Department of Computer Science
Dalhousie University
Halifax, Canada
hammad.rizwan@dal.ca

Hassan Sajjad
Department of Computer Science
Dalhousie University
Halifax, Canada

Peizhong Ju
Department of Computer Science
University of Kentucky
Kentucky, USA

A.B. Siddique
Department of Computer Science
University of Kentucky
Kentucky, USA

Abstract

Interpreting the internal mechanisms of large language models (LLMs) is crucial for improving their trustworthiness and utility. Prior work has primarily focused on mapping individual neurons to discrete semantic concepts. However, such mappings struggle to handle the inherent polysemanticity in LLMs, where individual neurons encode multiple, distinct concepts. Through a comprehensive analysis of both encoder and decoder-based LLMs across diverse datasets, we observe that even highly salient neurons, identified via various attribution techniques for specific semantic concepts, consistently exhibit polysemantic behavior. Importantly, activation magnitudes for fine-grained concepts follow distinct, often Gaussian-like distributions with minimal overlap. This observation motivates a shift from neuron attribution to range-based interpretation. We hypothesize that interpreting and manipulating neuron activation ranges would enable more precise interpretability and targeted interventions in LLMs. To validate our hypothesis, we introduce NeuronLens, a novel range-based interpretation and manipulation framework that provides a finer view of neuron activation distributions to localize concept attribution within a neuron. Extensive empirical evaluations demonstrate that NeuronLens significantly reduces unintended interference, maintaining precise manipulation of targeted concepts, outperforming neuron attribution.

1 Introduction

Large language models (LLMs) demonstrate remarkable performance across natural language understanding, generation, and transformation tasks [Brown et al., 2020, Bommasani et al., 2022, Touvron et al., 2023, Raffel et al., 2019]. However, the inner workings of LLMs remain largely

*Corresponding author: muhammadumairhaider@uky.edu
†Corresponding author: hammad.rizwan@dal.ca

Mechanistic Interpretability Workshop at NeurIPS 2025

opaque [Burkart and Huber, 2021], as their representations are distributed across billions of parameters. This lack of interpretability raises critical concerns about reliability, fairness, and trustworthiness, particularly in high-stakes domains such as healthcare, law, and education. To this end, neuron-level interpretability can address these concerns by enabling researchers to uncover how individual neurons encode semantic concepts and contribute to model outputs. With this understanding, researchers can diagnose safety risks [Wei et al., 2024, He et al., 2024], refine model outputs [Meng et al., 2023, Rizwan et al., 2024], optimize efficiency through pruning [Frankle and Carbin, 2019, Haider and Taj, 2021], and steer model's representations toward desired objectives [Subramani et al., 2022, Li et al., 2023, Rodriguez et al., 2024].

Recent research efforts have made significant progress in neuron-level interpretation by identifying salient neurons that influence model behavior [Dalvi et al., 2019a, Antverg and Belinkov, 2022, Conmy et al., 2023, Marks et al., 2024]. Approaches such as maximal activation analysis [Foote et al., 2023, Frankle and Carbin, 2019], Probe-based methods that employ auxiliary classifiers to distinguish between concepts [Dalvi et al., 2019a,b], and the probeless approach bypasses the need for classifiers by directly analyzing neurons [Antverg and Belinkov, 2022]. Other techniques include circuit discovery, attributes concepts to groups of interacting neurons [Olah et al., 2020, Conmy et al., 2023], and causal analysis, identify the internal components role in model behavior [Vig et al., 2020].

While these methods have advanced our understanding of neurons, they often rely on discrete neuron-to-concept mappings, which assume that entire neurons encode single concepts. However, neurons frequently exhibit polysemanticity; the ability to encode multiple, seemingly unrelated concepts [Lecomte et al., 2024, Marshall and Kirchner, 2024]. Given this heterogeneous encoding of concepts, traditional approaches often lead to unintended consequences when manipulating neurons, as changes intended for one concept may inadvertently affect others encoded by the same neuron or suboptimal interpretations of concepts [Sajjad et al., 2022].

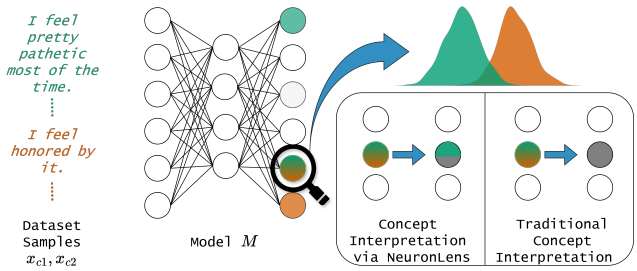

Figure 1: NeuronLens leverages distinct, Gaussian-like activation patterns to enable fine-grained concept attribution.

Despite being traditionally viewed as a challenge, could polysemanticity instead provide a unique lens for advancing interpretability and model control? If individual neurons encode multiple concepts, might their activation spectrum reveal distinct and identifiable patterns for each concept? Could these patterns enable precise interventions that adjust one concept while minimizing interference with others, overcoming the limitations of coarse, monolithic neuron-to-concept mappings?

This work seeks to address these questions by analyzing the activation patterns of neurons in both encoder-based and decoder-based LLMs. Through statistical and qualitative analysis, we find that neuronal activations for concepts follow *Gaussian-like distributions*, with distinct patterns for different concepts. Our key insight is that the unit of interpretability lies at a level more fine-grained than the neuron itself. Within a neuron's activation spectrum, *activation ranges corresponding to specific concepts can be used as a finer unit of interpretability*. This nuanced perspective enables a more precise approach to neuron interpretation and manipulation, addressing the limitations of traditional, discrete neuron-to-concept mappings.

Building upon these insights, we introduce NeuronLens visualised in Figure 1, a range-based framework for neuron interpretation and manipulation. NeuronLens identifies and maps specific activation ranges within a neuron's distribution to individual concepts, rather than attributing entire neurons to single concepts. For each concept, NeuronLens calculates a range that covers its activation spectrum, capturing the concept-specific activations while excluding unrelated concepts. Through experiments on encoder-based and decoder-based LLMs across several text classification datasets, we show that NeuronLens significantly reduces unintended interference by up to 25 percentage points in auxiliary concepts and up to 7x in LLM, while maintaining precise manipulation of targeted concepts, outperforming existing methods.

Our key contributions are: (1) To the best of our knowledge, this is the first work that performs a comprehensive study unfolding polysemantic neurons using activation spectrums. (2) We show that

Table 1: Performance drops relative to Baseline configuration (i.e., unaltered model's performance) for three techniques: Probeless, Probe, and Max. All values show the difference from Base values. Results are for *Emotions* dataset on the GPT-2 model using 30% salient neurons of each method. Metrics are detailed in Section 2.1.

Probeless				Probe				Max			
Acc	**Conf**	**CAcc**	**CConf**	**Acc**	**Conf**	**CAcc**	**CConf**	**Acc**	**Conf**	**CAcc**	**CConf**
-0.524	-0.510	-0.086	-0.086	-0.052	-0.036	-0.018	-0.049	**-0.735**	**-0.739**	-0.103	-0.103

neuronal activations in LLMs form distinct **concept level** Gaussian-like distributions, with salient neurons exhibiting limited overlap in their activation patterns across concepts. (3) We empirically demonstrate that activation ranges within a neuron's activation spectrum offer a more precise unit of interpretability, offering a refined framework for neuron-level analysis. (4) We propose NeuronLens, an activation range-based framework for interpreting and manipulating neuronal activations, which enables fine-grained concept attribution and reduces unintended interference compared to neuron level intervention.

2 Neuron Interpretation Analysis

This section provides an overview of the neuron analysis, methods for extracting salient neurons, and causally validating their saliency.

2.1 Preliminaries

Neuron. We refer to the output of an activation as a neuron. In a transformer model, we consider neurons of hidden state vectors of different transformer layers. Formally, given a hidden state vector $\mathbf{h}^l \in \mathbb{R}^d$ of size d produced by layer l, h^l_j denotes its j-th neuron, i.e., the j-th component of $\mathbf{h}^l$.

Concept. A concept $c \in C$ is a high-level semantic category that groups each input instance (or components of every instance), where C is the set of all concepts. For example, in a language task, a sentence can be categorized into 4 types: declarative, interrogative, imperative, and exclamatory, where each type is a concept. Words of a sentence can also have concepts like nouns, verbs, adjectives, adverbs, etc. In this study, we focus on the situation where all input samples are labelled with concepts.

Saliency Ranking. A saliency ranking orders the importance of neurons based on some saliency metric. For a hidden state vector $\mathbf{h}^l \in \mathbb{R}^d$, $s_{j,c}$ denotes the value of the saliency metric for the j-th neuron with respect to a concept c. The saliency ranking $(r_c(1), r_c(2), \cdots, r_c(d))$ is a permutation of the indices of neurons $(1, 2, \cdots, d)$, where $r_c(j) < r_c(i)$ if $s_{j,c} > s_{i,c}$. The saliency metric is usually predetermined, e.g., the absolute value of each neuron.

Concept Learning. Given a hidden state vector $\mathbf{h}^l$ as input, the associated concept can be the output of an appended neural network (e.g., several fully connected layers). The parameters of this appended neural network can be trained using training samples labelled with concepts.

Metrics. To establish the causal validity of the attribution, we employ two quantitative metrics: prediction accuracy and the model's predictive probability as a proxy for confidence score. First, baseline measurements of both accuracy and confidence for all concepts C without any intervention (unmodified model) are established. Post-intervention measurements are recorded for the target concept c and auxiliary concepts(other concepts in the dataset) c'. The effectiveness and precision of attribution are assessed through two key metrics: (1) the magnitude of performance degradation for concept c, and (2) the extent of unintended impact on auxiliary concepts c'. Throughout our analysis, we denote the accuracy and confidence metrics for concept c as **Acc** and **Conf** respectively, while corresponding measurements for auxiliary concepts c' are represented as **CAcc** and **CConf**. For evaluating the effect of the interventions on LLMs latent capabilities, we utilize **perplexity (PPL)** and **MMLU** [Hendrycks et al., 2021] zero-shot accuracy.

2.2 Concept Erasure

To assess the performance of a neuronal attribution, concept erasure acts as a critical diagnostic intervention to determine the causal effect of identified neurons for a given concept [Dalvi et al.,

2019b, Dai et al., 2022, Dalvi et al., 2019c, Morcos et al., 2018]. The core idea is that if a neuron is salient to a concept, eliminating it should result in the degradation of that concept's performance while causing minimal disruption to other concepts. This can be formalized as follows: given a concept-learning model M that maps any input instance x (or part of an instance) to a concept $M(x) = c \quad c \in C$, an ideal intervened model M'_{ideal} after erasing a target concept $c \in C$ should satisfy the following property:

$$M'_{\text{ideal}}(x) = \begin{cases} \neq M(x) & \text{if } M(x) = c, \\ = M(x) & \text{if } M(x) \neq c. \end{cases}$$

A popular approach of concept erasure in neuronal analysis literature [Dai et al., 2022, Antverg and Belinkov, 2022] is zeroing out specific neurons that are "important" to the target concept. Other studies have argued that zeroing out neurons is an overly aggressive intervention that can lead to catastrophic degradation in model performance. In Appendix Section D, we provide an ablation comparing different activation interventions for concept erasure.

2.3 Salient Neurons Extraction

Problem setup and preparation: We record activations for training samples of different concepts to perform neuron interpretation. Specifically, if we want to interpret neurons of $\mathbf{h}^l$ (hidden state vector at layer l), we traverse the training dataset and store the values of $\mathbf{h}^l$ and the associated concepts of all samples into a set H^l. The set H^l is further partitioned into H^l_c for all concepts $c \in C$. Such preparation is common in the relevant literature [Dalvi et al., 2019c,b, Antverg and Belinkov, 2022].

Max activations. Frankle and Carbin [2019] extract high activations as a saliency ranking metric relying upon the rationale that maximally activating neurons with respect to a concept c are important for that concept. **Probe analysis.** Dalvi et al. [2019b] train a linear classifier on the hidden representations H^l_c to predict each concept. The learned model weights are then utilized as a saliency ranking. **Probeless.** Antverg and Belinkov [2022] examine individual neurons, without the need for auxiliary classifiers, using the element-wise difference between mean vectors. Details of these approaches are provided in Appendix G.

To assess the effectiveness of the attribution methods, we perform neuron masking on the salient neurons identified by each method in a concept erasure task. Table 1 provides the results for this experiment. We observe that irrespective of the method used to obtain saliency ranking, a single concept eraser using salient neurons causes deterioration in performance across several concepts. Max activation causes the highest degree of deterioration in the targeted concept while maintaining a comparable deterioration in auxiliary concepts. Based on this finding, we adopt max activation ranking for saliency ranking. Moreover, we hypothesize that one reason for such deterioration in overall performance is due to the polysemantic nature of neurons.

3 Polysemanticity

The polysemanticity of neuronal units, including salient neurons that encode information about multiple concepts, poses a challenge to neural network interpretation and manipulation. In this section, we discuss the degree of polysemanticity in salient neurons in detail.

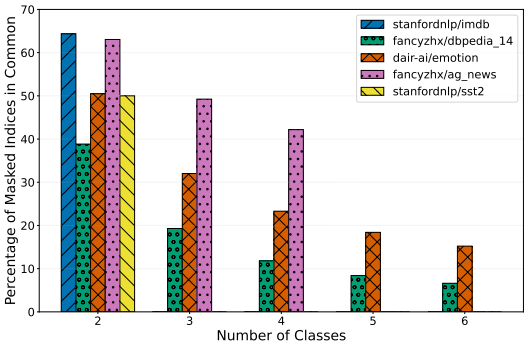

Figure 2: Overlap of top 30% salient neurons across classes.

Polysemanticity often arises when models must represent more features than their capacity allows or due to specific training paradigms. Limited representational space forces neurons to encode multiple unrelated features to maintain performance [Anthropic, 2023]. Training methods like subword tokenization, designed to reduce vocabulary size and model complexity, lead to context-dependent token splits, causing neural activations to encode multiple meanings [Sennrich et al., 2016, Elhage et al., 2022, Meng et al.,

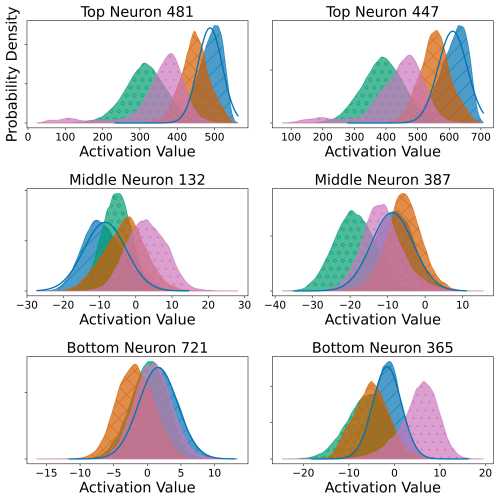

Figure 3: Neuronal Activation Patterns of six neurons on *AG-News* dataset *class 1*. Neurons 418 and 447 are the highest activating neurons, neurons 132 and 387 are middle-ranked neurons, and neurons 721 and 365 are the lowest activating neurons.

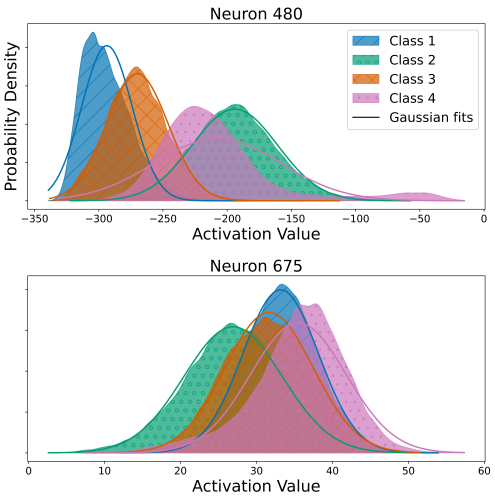

Figure 4: Comparison of neurons 480 and 675 showing class-specific activation patterns and fitted Gaussian curves. Both neurons were salient across all classes in top 5% on *AG-News*.

2022]. Additionally, Lecomte et al. [2024] show that even with sufficient capacity, certain weight initializations can induce polysemanticity by placing neurons near multiple conceptual regions.

Polysemanticity in salient neurons. Given that salient neurons have a strong causal association with the concept of interest, their tendency to be mono-semantic should be is high, but we find that there is a high degree of polysemanticity in salient neurons. We investigate this by extracting 30% salient neurons (i.e.: Max neurons) for different datasets on the GPT-2 model. The results show that there is a considerable overlap of salient neurons between concepts (classes) as shown in Figure 2. In the case of a two-class dataset IMDB, the overlap of salient neurons, selected by max, is more than 60%. This shows a high degree of polysemanticity. Consequently, we extrapolate that salient neural representations may exist in a polysemantic configuration, wherein a subset of the salient neurons encode information through intricate activation patterns.

The monolithic attribution paradigm potentially oversimplifies the complex, distributed nature of neuronal activation as can be seen in polysemanticity [Lecomte et al., 2024, Marshall and Kirchner, 2024] where a single neuron learns multiple seemingly unrelated concepts and elucidates them at different activation values.

4 Neuronal Activation Patterns

In this section, we analyze the properties of neuronal activations of the salient neurons (including polysemantic) extracted via maximal activation. Similar to [Gurnee et al., 2024], our findings indicate that neuronal activations form a **Gaussian-like distribution**. We further find that salient neurons have a **distinct Gaussian distribution of activations for different concepts** with limited overlap with other concept activations.

Qualitative Evaluation. To visually demonstrate that neuron activations for a concept c follows a Gaussian-like distribution, we extract model representations as described in Section 2.1. Using saliency ranking r_c for a single concept class, we examine neurons from different ranking positions in the GPT-2 model on the AG-News dataset: two top-ranked neurons ($r_c \leq 2$), two middle-ranked neurons ($r_c \approx d/2$), and two bottom-ranked neurons ($r_c \geq d - 1$). In Figures 3 and 4, we use Kernel Density Estimation (KDE) to visualize these distributions. Figure 3 reveals that while the activations are Gaussian-like for different concepts, salient neurons demonstrate distinct activation patterns with limited overlap, middle-ranked neurons show a higher degree of overlap than the top ones, whereas non-salient neurons (bottom two) exhibit the highest overlap in their activation distributions.

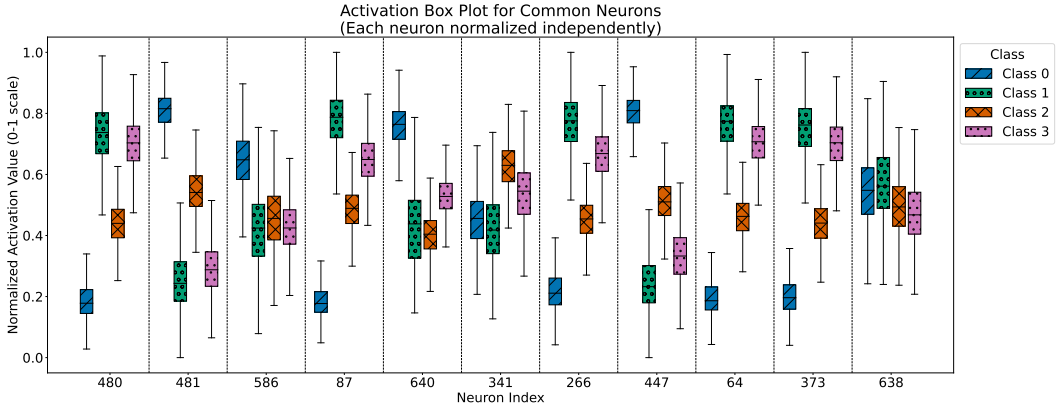

Figure 5: Box plot of neural activation of 11 polysemantic neurons (i.e: neurons in the salient group for all classes, percentage selected: 5% top salient) for 4 randomly selected classes out of 14 classes of *DBPedia-14* dataset.

Additionally in Figure 4, we identify and visualize two distinct types of polysemantic neurons that appear in the salient sets across all classes, when 5% salient set was selected, in the dataset. The first type, exemplified by neuron 480, maintains partially separable activation patterns despite being polysemantic, suggesting some degree of class-specific behavior. In contrast, the second type, represented by neuron 675, exhibits completely overlapping activation patterns across all classes, making it hard to disentangle. To further investigate this phenomenon, Figure 5 presents a broader analysis of neurons from the polysemantic subset, identified using a 5% saliency threshold(top 5% salient neurons selected for a concept). By examining these neurons' behavior across four randomly selected classes (out of 14 total classes), we observe that most polysemantic neurons exhibit a high degree of separability, for some classes, while they respond to multiple classes, they tend to operate in partially separable activation ranges, supporting the possibility of meaningful disentanglement.

Quantitative Evaluation. To quantify the effect of Gaussian-like distribution of neurons for $c \in C$, we perform statistical analysis of activations. We computed the skewness, kurtosis [Joanes and Gill, 1998] and analyzed the normality of neuronal activations using Kolmogorov-Smirnov (KS) test [Massey Jr, 1951]. Table 2 presents the results for distributional properties across all neurons. The average skewness is close to 0 across all datasets, indicating strong symmetry (ideal normal distribution: 0), and the average kurtosis is close to 3, nearly identical to the expected value for a normal distribution (3.0).

Table 2: Skewness, kurtosis, and Kolmogorov-Smirnov test results across various datasets.*GPT-2* model

Dataset	Skewness	Kurtosis	KS-Test
stanfordnlp/imdb	0.0014	3.6639	1.0000
fancyzhx/dbpedia_14	-0.0007	3.9360	0.9782
dair-ai/emotion	0.0015	3.0198	0.9446
fancyzhx/ag_news	-0.0013	3.2060	0.9918
stanfordnlp/sst2	-0.0083	3.2038	1.0000

To quantitatively assess normality, while accounting for practical significance, we employ the KS test with an effect size threshold of 10%. This approach tests whether the distribution remains within a reasonable bound of normality, rather than testing for perfect normality, which is overly strict for real-world data. For each neuron, we normalize the activations to zero mean and unit variance, then compute the KS statistic against a standard normal distribution. The KS statistic represents the maximum absolute difference between the empirical and theoretical cumulative distribution functions. Using a threshold of 0.1 (allowing a maximum 10% deviation from normal), we find that close to 100% of the neurons exhibit practically normal distributions. The combination of near-ideal skewness and kurtosis values, visual confirmation through KDEs, and our effect size-based KS tests provide strong evidence that the activations follow approximately normal distributions.

We report quantitative statistics for all layers in Appendix Section I.1, which show that as layer depth increases, kurtosis steadily converges toward the Gaussian benchmark of 3.0, skewness remains near zero, and the 10% practical-normality score stays close to 1 across the network. A qualitative, layer-wise examination in Appendix Section I.2 further reveals that while all layers exhibit class level Gaussian-like activations. Early layers show substantial overlap between classes, this is consistent with the understanding that earlier layers focus on low-level features, not high level features like class.

Beginning as early as layers 5–6, distinct class-specific Gaussians emerge and become progressively more separable in deeper layers, indicating a transition toward higher-level semantic representations.

5 Activation Ranges-guided Concept Erasure

Given that neuronal activations exhibit approximately Gaussian-like distributions with separable means, we can interpret and intervene on neurons more precisely than by ablating entire units. Specifically, NeuronLens ablates salient neurons identified through saliency ranking only when their activation falls within a selected range. The key idea is to identify a range that is strongly associated with the target concept c intended for erasure. This range-based approach enables fine-grained ablation, thereby reducing unintended interference with non-target concepts. To validate our approach, we evaluate the causal efficacy of our method relative to neuron ablation using concept erasure experiments and assess the model's latent capabilities following this intervention.

To calculate the aforementioned range, the framework utilises the means and standard deviations of the neuron activations. Specifically, first the empirical average $\mu \in \mathbb{R}$ and standard deviation $\sigma \geq 0$ of the values of the salient neuron for all samples associated with the target concept $c \in C$ are calculated. After that, range is assigned as $[\mu - \tau \times \sigma,\ \mu + \tau \times \sigma]$, where $\tau > 0$ is a hyperparameter to make a tradeoff between erasing the target concept c (using larger τ) and smaller impact on auxiliary concepts and general LLM capabilities (using smaller τ). For this work, τ is set to $\tau = 2.5$, assuming a fully Gaussian distribution. This threshold corresponds to a coverage of approximately 98.76% of the distribution's values, providing a slightly conservative bound for range-based interventions. Ablations for varying the hyperparameter τ are presented in Appendix Section H, the results indicate that targeted concept deteriorates up to 2.4-2.7 τ then plateaus, while auxiliary concepts begin to degrade further.

$$h_j^l(x) = \begin{cases} \phi(x) & \text{if } h_j^l(x) \in \text{CR}(l,j,c) \\ h_j^l(x) & \text{otherwise} \end{cases} \qquad \begin{aligned} &\text{CR}(l,j,c) = [\mu - 2.5\sigma,\ \mu + 2.5\sigma], \\ &\mu = \frac{1}{|H_c^l|} \sum_{\mathbf{h}^l \in H_c^l} h_j^l, \sigma = \sqrt{\frac{1}{|H_c^l|} \sum_{\mathbf{h}^l \in H_c^l} (h_j^l - \mu)^2} \end{aligned}$$

where CR represents Correlated Range and $\phi()$ is the activation intervention function, which returns zero for the results presented in the main paper.

Notice that H_c^l was defined in problem setup and preparation of section 2.3, which denotes the set of hidden state vector $\mathbf{h}^l(x_c)$ at layer l for all training samples x_c associated with concept c. Here $|\cdot|$ denotes the cardinality of a set.

5.1 Experimental Setup

Models. This study employs both encoder and decoder-based models, including **fine-tuned** BERT [Devlin et al., 2019], DistilBERT [Sanh et al., 2020], GPT-2 [Radford et al., 2019], and **pretrained** Llama-3.2-3B [Grattafiori, 2024]. We incorporate our methodology at the penultimate layer; ablation for layer selection is provided in the Appendix Section I. The training details for the models are provided in Appendix Section E.

For trained models (BERT, DistilBERT, and GPT-2), a higher proportion of neurons (up to 50%) can be ablated with a relatively minor impact on primary task performance and minimal interference with auxiliary concepts. This suggests substantial neuronal redundancy, wherein multiple neurons appear to encode overlapping features.

Datasets. We consider various classification based tasks; sentiment analysis (IMDB, [Maas et al., 2011]), (SST2, [Socher et al., 2013]), emotion detection (Dair-Ai/Emotions Saravia et al. [2018]), news classification (AG-News [Zhang et al., 2015]) and article content categorization (DBPedia-14 [Zhang et al., 2015]).

Table 3: Evaluation of selected models on IMDB, SST2, AG-News, and DBPedia-14 datasets using activation range and neuron masking techniques. Performance metrics are calculated using class level 10% trimmed means at the class level. Metrics are detailed in Section 2.1. For *GPT-2* and *Bert* 50% and for *Llama-3.2-3B* 30% neurons selected.

Model	Dataset	Base Values				Neuron Masking				Activation Range Masking			
		Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf
BERT	IMDB	0.928	0.904	0.928	0.904	-0.190	-0.353	0.059	-0.078	-0.184	-0.360	0.058	**0.030**
	SST2	0.910	0.903	0.910	0.903	-0.051	-0.313	0.031	-0.046	-0.060	-0.330	0.031	**0.043**
	AG-NEWS	0.948	0.929	0.948	0.929	-0.271	-0.590	0.012	-0.074	-0.261	-0.590	**0.013**	**-0.009**
	Emotions	0.894	0.834	0.917	0.876	-0.291	-0.633	0.013	-0.265	-0.279	-0.635	**0.014**	**-0.069**
	DBPedia-14	0.992	0.991	0.990	0.989	-0.028	-0.786	0.000	-0.017	-0.015	-0.766	0.000	**-0.000**
GPT-2	IMDB	0.952	0.939	0.952	0.939	-0.196	-0.188	**0.033**	**0.045**	-0.195	-0.197	0.031	0.042
	SST2	0.966	0.958	0.966	0.958	-0.165	-0.190	0.025	0.032	-0.159	-0.192	0.025	**0.028**
	AG-NEWS	0.945	0.933	0.945	0.933	-0.871	-0.877	-0.155	**-0.163**	-0.849	-0.862	**-0.063**	-0.223
	Emotions	0.905	0.892	0.930	0.919	-0.735	-0.738	-0.103	-0.103	-0.737	-0.739	**-0.044**	**-0.046**
	DBPedia-14	0.993	0.990	0.990	0.988	-0.810	-0.845	-0.154	-0.177	-0.782	-0.825	**-0.015**	**-0.031**
Llama	IMDB	0.952	0.939	0.952	0.939	-0.196	-0.188	0.033	0.045	-0.195	-0.197	**0.031**	**0.042**
	SST2	1.000	0.559	1.000	0.559	-0.760	-0.429	-0.394	-0.295	-0.756	-0.427	**-0.384**	**-0.291**
	AG-NEWS	1.000	0.744	1.000	0.744	-0.934	-0.725	-0.660	-0.572	-0.935	-0.725	**-0.484**	**-0.454**
	Emotions	0.815	0.472	0.823	0.477	-0.795	-0.470	-0.696	-0.429	-0.797	-0.469	**-0.594**	**-0.404**
	DBPedia-14	1.000	0.533	1.000	0.563	-0.992	-0.528	-0.912	-0.445	-0.986	-0.527	**-0.663**	**-0.354**

5.2 Results and Analysis

Table 3 presents results for the concept removal task across five benchmark datasets (Class-wise detailed results are provided in Appendix Section J), demonstrating the effectiveness of our range-based masking approach compared to traditional neuron masking.

On binary classification tasks (IMDB, SST2), both masking approaches show moderate performance drops in targeted concepts. This suggests higher redundancy for coarser binary concepts. Multi-class classification tasks with fine-grained labels, such as AG-News, Emotions, and DBPedia-14, exhibit more pronounced effects under intervention. Range-based masking results in significant degradation of primary task performance while preserving auxiliary concept accuracy, this is particularly evident in results for AG-News.

GPT-2, despite being fine-tuned but trained in an autoregressive manner, shows substantially higher vulnerability with major drops in AG-NEWS ($\Delta_{acc} = -0.849$) and DBPedia-14 ($\Delta_{acc} = -0.782$). This increased sensitivity may be attributed to its autoregressive training objective, which potentially leads to more sequential and less redundant concept encodings. The Llama-3.2-3B model, evaluated in a few-shot setting without task-specific training, experiences the most severe degradation across all datasets (often exceeding -0.90), suggesting that pre-trained representations without task-specific fine-tuning are more vulnerable to targeted neuron interventions.

Table 4 in Appendix Section C presents the results showing the impact of concept erasure intervention on latent LLM capabilities such as fluency and generalization. Neuron masking degrades performance, increasing perplexity by (3.8-5.74) and lowering MMLU accuracy. In contrast, activation range masking raises perplexity by (0.5-1.1) points only, while preserving or improving MMLU scores indicates more precise and less disruptive removal.

Alternative activation interventions, beyond zeroing out, are explored in Appendix Section D, including the *dampening* method [Suau et al., 2024] and *mean replacement* [Suau et al., 2021]. While these methods aim to manipulate without moving too far from the original representation, they exhibit limitations when applied to neurons. Specifically, neuron dampening increases perplexity by 2.9–3.7 points and often degrades MMLU accuracy (up to -0.045), whereas range-based dampening confines perplexity increases to 0.5–0.8 points and occasionally improves MMLU (up to +0.035). Similarly, mean replacement leads to substantial degradation when applied to neurons (perplexity increases of 7.4–8.8), while range-restricted replacement reduces the impact to below 0.7 points.

However, all approaches suffer from rigid static suppression or substitution, which fail to account for concept-specific activation dynamics. To address this issue, we introduce a novel **adaptive dampening** technique. This method modulates suppression in proportion to each activation's deviation from its class-conditional mean, enabling data driven suppression. Adaptive dampening achieves the strongest balance across all metrics: perplexity remains low (0.41–0.61), MMLU is maintained or improved (up to +0.03), and collateral damage to auxiliary concepts is minimized (CAcc

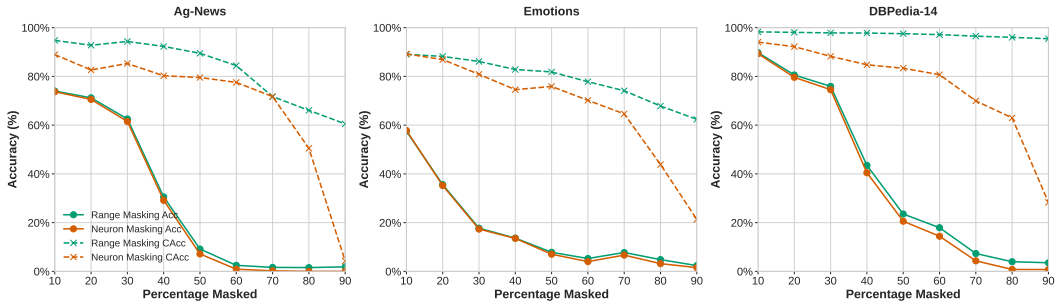

Figure 6: Accuracy comparisons between Neuronal Range manipulation (green) and complete neuron manipulation (orange) methods on *GPT-2* model.

drops consistently below -0.3, often under -0.15), outperforming dampening, mean replacement and zeroing out approaches.

These results demonstrate that precise intervention in specific activation ranges, enables significantly more targeted concept manipulation while preserving auxiliary concepts, highlighting how conceptual information is encoded within specific activation patterns rather than isolated to individual neurons and underscoring the importance of activation ranges in capturing neuron-concept relationships.

Percentage Masking Effect

As more neurons are masked, performance gains of range-based masking over the neuron masking baseline become increasingly evident. Beyond a critical threshold of the number of masked neurons, baseline performance degrades sharply, while our method remains stable up to masking of 100% neurons. This arises from two factors: (1) models have a large number of polysemantic neurons and higher masking rates increase the chance of ablating them, and (2) blocking/manipulating a higher percentage of the model's neurons creates a significant deviation from the original model's behavior. For low-activation neurons with respect to the concept of interest, discrete neuron masking i.e. completely masking out a neuron, becomes unreliable, as shown in Figure 6, with a steep performance drop after masking 50% neurons. This underscores the need for finer-grained attribution; our range-based method offers such precision, preserving model behaviour under extensive masking.

The relatively stable results on auxiliary concept when using range-based masking at high percentage of neurons reduces the need to find an optimum threshold for the number of neurons to ablate which is critical to neuron masking.

6 Related Work

While we have discussed closely related approaches in Section 2, here, we briefly review additional relevant techniques. Circuit discovery identifies groups of neurons that jointly encode concepts, providing a structured view of model behavior [Marks et al., 2024, Conmy et al., 2023, Olah et al., 2020]. However, extracting circuits is computationally intensive and lacks fine-grained neuron-level attribution. Gradient-based methods attribute predictions to input features by tracking gradients through the network, with integrated gradients [Sundararajan et al., 2017, Dai et al., 2022] being a widely used approach. However, they struggle with polysemanticity, as they do not disentangle overlapping concepts within neurons. Causal analysis methods intervene on internal components to assess their role in encoding concepts. Causal tracing measures the effect of corrupting activations on model performance [Vig et al., 2020, Meng et al., 2022], while causal mediation analysis quantifies information propagation through neurons [Vig et al., 2020]. Although effective, these techniques require costly perturbation experiments. Beyond neuron-level analysis, representation-level methods examine hidden states and their relationship to model outputs and concepts [Veldhoen et al., 2016, Tenney et al., 2019, Liu et al., 2019]. Sparse probing [Gurnee et al., 2023] compresses hidden representations into sparse, interpretable subspaces. While prior work has advanced interpretability, most methods rely on discrete neuron-to-concept mappings, which fail to account for polysemanticity [Sajjad et al., 2022]. Our work extends activation-based approaches by introducing activation ranges as the unit of interpretability to enable more precise concept attribution and intervention.

7 Conclusion

In this work, we challenged traditional assumptions about neuron interpretability by reframing polysemanticity as a resource rather than a limitation in interpreting neurons. Through an in depth analysis, we uncovered that neuronal activations for individual concepts exhibit distinct, Gaussian-like distributions. This discovery allows for a more precise understanding of how neurons encode multiple concepts, enabling us to move beyond coarse, monolithic neuron-to-concept mappings. Building upon these insights, we proposed NeuronLens, a novel range-based framework for neuron interpretation and manipulation. NeuronLens offers fine-grained control that reduces interference with unrelated concepts by attributing specific activation ranges within neurons to individual concepts. Extensive empirical evaluations demonstrated that NeuronLens outperforms neuronal attribution methods in maintaining concept-specific precision while minimizing unintended side effects. Notably, while targeted concept removal remains equally effective when comparing neuron vs range based interventions, our approach achieves superior preservation of auxiliary concepts without compromising the primary goal. An important direction for future work is exploring our range-based method as a metric for quantifying polysemanticity in neural networks. This approach may also serve as a diagnostic tool to evaluate the effectiveness of sparse autoencoders (SAEs) in disentangling concept representations across individual neurons.

References

Anthropic. A toy model of double descent from sparsely-gated routing. *Transformer Circuits*, 2023. URL https://transformer-circuits.pub/2023/toy-double-descent/index.html.

Omer Antverg and Yonatan Belinkov. On the pitfalls of analyzing individual neurons in language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=8uzOEWPQIMu.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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, 2020.

Nadia Burkart and Marco F. Huber. A survey on the explainability of supervised machine learning. *J. Artif. Intell. Res.*, 70:245–317, 2021. doi: 10.1613/JAIR.1.12228. URL https://doi.org/10.1613/jair.1.12228.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36:16318–16352, 2023.

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, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.581. URL https://aclanthology.org/2022.acl-long.581/.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James R. Glass. What is one grain of sand in the desert? analyzing individual neurons in deep NLP models. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6309–6317. AAAI Press, 2019a. doi: 10.1609/AAAI.V33I01.33016309. URL https://doi.org/10.1609/aaai.v33i01.33016309.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, D. Anthony Bau, and James Glass. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, March 2019b.

Fahim Dalvi, Avery Nortonsmith, D Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, and James Glass. Neurox: A toolkit for analyzing individual neurons in neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019c.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Nelson Elhage et al. Superposition, memorization, and double descent. *Transformer Circuits*, 2022.

Alex Foote, Neel Nanda, Esben Kran, Ionnis Konstas, and Fazl Barez. N2g: A scalable approach for quantifying interpretable neuron representations in large language models. *arXiv preprint arXiv:2304.12918*, 2023.

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

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks, 2019. URL https://arxiv.org/abs/1803.03635.

Aaron Grattafiori. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing, 2023. URL https://arxiv.org/abs/2305.01610.

Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. Universal neurons in gpt2 language models. *arXiv preprint arXiv:2401.12181*, 2024.

Muhammad Umair Haider and Murtaza Taj. Comprehensive online network pruning via learnable scaling factors. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3557–3561, 2021. doi: 10.1109/ICIP42928.2021.9506252.

Zeqing He, Zhibo Wang, Zhixuan Chu, Huiyu Xu, Rui Zheng, Kui Ren, and Chun Chen. Jailbreaklens: Interpreting jailbreak mechanism in the lens of representation and circuit. *arXiv preprint arXiv:2411.11114*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Derrick N Joanes and Christine A Gill. Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):183–189, 1998.

Victor Lecomte, Kushal Thaman, Rylan Schaeffer, Naomi Bashkansky, Trevor Chow, and Sanmi Koyejo. What causes polysemanticity? an alternative origin story of mixed selectivity from incidental causes. In *ICLR 2024 Workshop on Representational Alignment*, 2024.

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

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1112. URL https://aclanthology.org/N19-1112/.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

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. *CoRR*, abs/2403.19647, 2024. doi: 10.48550/ARXIV.2403.19647. URL https://doi.org/10.48550/arXiv.2403.19647.

Simon C. Marshall and Jan H. Kirchner. Understanding polysemanticity in neural networks through coding theory, 2024. URL https://arxiv.org/abs/2401.17975.

Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 2022.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Ari S. Morcos, David G. T. Barrett, Neil C. Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization, 2018. URL https://arxiv.org/abs/1803.06959.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Hammad Rizwan, Domenic Rosati, Ga Wu, and Hassan Sajjad. Resolving lexical bias in edit scoping with projector editor networks. *arXiv preprint arXiv:2408.10411*, 2024.

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Marco Cuturi, and Xavier Suau. Controlling language and diffusion models by transporting activations. *arXiv preprint arXiv:2410.23054*, 2024.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. Neuron-level interpretation of deep nlp models: A survey, 2022. URL https://arxiv.org/abs/2108.13138.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL https://arxiv.org/abs/1910.01108.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1404. URL https://www.aclweb.org/anthology/D18-1404.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

Xavier Suau, Luca Zappella, and Nicholas Apostoloff. Self-conditioning pre-trained language models. *arXiv preprint arXiv:2110.02802*, 2021.

Xavier Suau, Pieter Delobelle, Katherine Metcalf, Armand Joulin, Nicholas Apostoloff, Luca Zappella, and Pau Rodríguez. Whispering experts: Neural interventions for toxicity mitigation in language models. *arXiv preprint arXiv:2407.12824*, 2024.

Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. Extracting latent steering vectors from pretrained language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 566–581. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-ACL.48. URL https://doi.org/10.18653/v1/2022.findings-acl.48.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017. URL https://arxiv.org/abs/1703.01365.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://www.aclweb.org/anthology/P19-1452.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Sara Veldhoen, Dieuwke Hupkes, and Willem Zuidema. Diagnostic classifiers: Revealing how neural networks process hierarchical structure. In *Pre-Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches (CoCo @ NIPS 2016)*, 2016.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. Neurons in large language models: Dead, n-gram, positional. *arXiv preprint arXiv:2309.04827*, 2023.

Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL `https://openreview.net/forum?id=K6xxnKN2gm`.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf`.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper claims that neuronal ranges are better units of interpretability and manipulation than neurons. The paper provides extensive ablations across encoder and decoder models to support this claim.

   Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitation are outlined in Section B.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: There are no theoretical proofs that require assumptions to be highlighted.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.

- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experimental setup is provided in Section 5.1 and training details are provided in Appendix Section E

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Datasets used are open source and are referenced in 5.1. Extensive experiment settings are provided in Appendix Section E. The code will be open-sourced upon paper acceptance.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Experimental setup is provided in Section 5.1 and training details are provided in Appendix Section E

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experiments were run once due to the significant computational cost associated with our setup (e.g., large model size, dataset scale, activations extraction, and model inferences). We do not report error bars or confidence intervals. We provide extensive experimentation across different settings, all of which support our claims.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute details are provided in Appendix Section F.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The proposed NeuronLens framework enables precise control of model behaviour, benefiting research for model safety and reliability. While this improved understanding could potentially be misused, the work's theoretical nature and focus on interpretability methods make immediate harmful applications are unlikely

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The proposed NeuronLens framework enables precise control of model behaviour, benefiting research for model safety and reliability highlighted in Section 1 of the main paper text. In Appendix Section A impact statement is provided.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release any new models or dataset. Neuronlens is a framework designed to understand the inner workings of large language models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The datasets and models used in the paper are referenced in Section 5.1

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes] .

    Justification: The paper provides a framework (NeuronLens) for understanding a large language model's internal workings. The methodology is defined in section 5.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing is performed for this work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: There are no human study participants for this work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [No]

Justification: LLM was only used to help with editing texts.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Impact Statement

This work advances neural network interpretability by providing a fine-grained understanding of concept encoding in language models. The proposed NeuronLens framework enables precise control of model behavior, benefiting research in model safety and reliability. While this improved understanding could potentially be misused, the work's theoretical nature and focus on interpretability methods makes immediate harmful applications unlikely.

B Limitations

While NeuronLens can disentangle polysemanticity to a degree using **Gaussian Like Distribution**, it is unable to completely disentangle concepts encoded in the polysemantic neurons, because there still is a significant overlap in the distributions of concepts in activations. Additionally, in this work, we use τ to be a fixed value of 2.5 to make the comparison of approaches fair, but τ selection can be optimized to be more sophisticated. We also get results primarily from the penultimate layer, and not the intermediate or earlier layers, however, we do give ablation and rationale for this choice in Appendix I

C General LLM Capabilities

We evaluate the general capabilities of large language models (LLMs) using the MMLU benchmark [Hendrycks et al., 2021] and perplexity scores on Wikipedia texts [Foundation]. Table 4 presents the comparative performance of neuron masking and activation range masking. Neuron masking leads to notable increases in perplexity, exceeding 3 points in the best case, whereas range masking results in a maximum increase of only 1.1. In terms of MMLU accuracy, neuron masking consistently reduces performance across all settings, while range masking preserves or improves performance in most cases, with degradation observed in only one instance.

Table 4: Evaluation of LLMs latent capabilities using Wikipedia for perplexity and zeroshot MMLU for testing generalisation capabilities. *Llama-3.2-3B model*

Dataset	Base Values		Neuron Masking		Activation Range Masking	
	Perplexity	MMLU	Perplexity	MMLU	Perplexity	MMLU
IMDB	7.007	0.530	10.990	0.515	7.550	0.530
SST2	7.007	0.530	11.688	0.510	8.150	0.537
AG-NEWS	7.007	0.530	12.757	0.510	8.022	0.533
Emotions	7.007	0.530	11.630	0.526	8.063	0.526
DBPedia-14	7.007	0.530	12.230	0.507	7.903	0.535

D Activation Intervention

In the main text, we primarily presented results using a "zeroing out" strategy for neuron manipulation. This approach was chosen to compare neuron manipulation against range-based manipulation.

However, zeroing out is considered a suboptimal strategy [Suau et al., 2024]. The primary concern with standard zeroing-out approaches is that they distort the activation distribution significantly, diverging from that of the original model. However, our range-based method selectively zeroes out only a narrow slice of the activation spectrum, thereby mitigating the adverse effects associated with hard erasure.

In this section, we explore alternative, more optimized strategies for concept removal. We also introduce a novel range-based scaling strategy that has demonstrated superior results.

Below, we explore various activation intervention strategies, comparing traditional neuron-level approaches with the nuanced range-based technique. Our comprehensive evaluation reveals that range-based manipulations consistently outperform neuron interventions across multiple metrics, with significantly less disruption to the model's general capabilities.

Among all techniques examined, our novel adaptive dampening approach emerges as the most effective, maintaining targeted concept suppression while minimizing collateral impact on auxiliary concepts and preserving overall language modelling capabilities. This pattern holds true across different intervention methods including zeroing out, dampening, and mean replacement strategies.

D.1 Dampening

In their work, Suau et al. [2024] propose using a dampening function rather than setting neuron activations to zero outright. This approach, referred to as DAMP, corresponds to a specific choice of the intervention function $\phi(x) = \alpha x$, where $0 \leq \alpha \leq 1$. In this formulation, the activations of selected neurons are scaled down by a factor α instead of being completely suppressed. Here, x represents neuron activation. The rationale behind dampening is that a fixed intervention (like zeroing out) can disrupt the LLM's inference dynamics, especially when a large number of neurons (k) are involved, thereby limiting its effectiveness. Dampening offers a less destructive intervention by allowing contextual signals to continue passing through the network. This, in turn, permits intervention on a larger set of expert neurons, potentially achieving stronger mitigation of the targeted concept.

Table 5: Evaluation of Llama-3.2-3B on a DBPedia-14 dataset using neuron and range masking techniques. 30% neurons were selected. Dampening factor used is $a = 0.125$. **Acc** represents class accuracy, **Conf** denotes class prediction probability, and **CAcc** and **CConf** refer to average accuracy and average class prediction probability across other classes, respectively. The *Base Values* indicate the baseline model performance, while *Neuron Masking* and *Activation Range Masking* show deviations from the baseline performance. PPL Δ and MMLU Δ show changes in perplexity and MMLU scores, respectively.

Class	Base Values				Neuron Masking (Deviations)						Activation Range Masking (Deviations)					
	Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf	PPL Δ	MMLU Δ	Acc	Conf	CAcc	CConf	PPL Δ	MMLU Δ
Class 0	1.000	0.576	1.000	0.563	-0.919	-0.545	-0.281	-0.309	3.161	-0.020	-0.924	-0.545	**-0.276**	**-0.285**	**0.640**	**-0.010**
Class 1	1.000	0.526	1.000	0.567	-0.988	-0.467	-0.246	-0.270	3.578	-0.015	-0.805	-0.466	**-0.193**	**-0.206**	**0.725**	**0.015**
Class 2	1.000	0.441	1.000	0.575	-0.864	-0.391	-0.461	-0.323	2.891	-0.030	-0.869	-0.389	**-0.346**	**-0.282**	**0.718**	**0.005**
Class 3	1.000	0.461	1.000	0.573	-0.974	-0.439	-0.411	-0.346	3.036	-0.025	-0.970	-0.438	**-0.282**	**-0.283**	**0.653**	**0.010**
Class 4	1.000	0.839	1.000	0.541	-0.382	-0.597	-0.367	-0.317	2.997	0.000	-0.382	-0.597	**-0.334**	**-0.284**	**0.691**	**0.020**
Class 5	1.000	0.339	1.000	0.568	-0.970	-0.326	-0.239	-0.246	3.503	0.010	-0.970	-0.325	**-0.197**	**-0.187**	**0.810**	**0.015**
Class 6	1.000	0.810	1.000	0.545	-0.233	-0.638	-0.194	-0.276	3.126	-0.010	-0.241	-0.637	**-0.174**	**-0.203**	**0.697**	-0.010
Class 7	1.000	0.595	1.000	0.562	-0.210	-0.382	-0.206	-0.226	3.037	0.000	-0.179	-0.376	**-0.123**	**-0.143**	**0.546**	**0.015**
Class 8	1.000	0.417	1.000	0.574	-0.310	-0.416	-0.335	-0.297	3.001	0.020	-0.346	-0.416	**-0.200**	**-0.187**	**0.624**	**0.015**
Class 9	1.000	0.526	1.000	0.567	-0.820	-0.465	-0.327	-0.264	3.369	-0.030	-0.809	-0.463	**-0.213**	**-0.189**	**0.596**	**0.000**
Class 10	1.000	0.505	1.000	0.569	-0.691	-0.466	-0.389	-0.314	3.732	**0.000**	-0.696	-0.465	**-0.267**	**-0.198**	**0.695**	-0.015
Class 11	1.000	0.497	1.000	0.569	-0.873	-0.432	-0.472	-0.289	3.070	-0.030	-0.865	-0.427	**-0.335**	**-0.205**	**0.594**	**-0.015**
Class 12	1.000	0.573	1.000	0.563	-0.720	-0.452	-0.295	-0.221	3.410	-0.045	-0.723	-0.451	**-0.190**	**-0.163**	**0.595**	**0.035**
Class 13	1.000	0.567	1.000	0.564	-0.951	-0.537	-0.226	-0.189	2.995	0.000	-0.955	-0.536	**-0.157**	**-0.150**	**0.672**	**0.005**

Table 5 presents a comparative analysis of two intervention strategies, neuron masking and activation range masking, when employing the Dampening technique with $\alpha = 0.5$. The evaluation spans 14 classes and utilizes the metrics: accuracy (Acc), confidence (Conf), class-wise accuracy (CAcc), class-wise confidence (CConf), alterations in perplexity (PPL), and MMLU score.

A consistent trend emerges across the primary metrics (Acc, Conf, CAcc, and CConf), where activation range masking demonstrates superior performance over neuron masking. Interventions based on activation ranges lead to a notably smaller decline in the accuracy and confidence associated with auxiliary concepts. For example, in Class 3, while neuron masking results in an accuracy drop of -0.974 in the targeted class and auxiliary class accuracy decrease of -0.411, activation range masking, despite a comparable accuracy reduction in the targeted class (-0.970), shows a less severe impact on

auxiliary class accuracy (-0.283). This pattern of activation range masking better preserves auxiliary class performance, is evident across all evaluated classes.

Examining the broader effects on language modeling capabilities reveals significant distinctions between the two approaches. Neuron masking results in a considerable rise in perplexity (PPL), with increases ranging from **+2.891 to +3.732** across all the classes. Furthermore, it tends to cause more pronounced negative shifts in MMLU scores, reaching as low as -0.045. Conversely, activation range masking results in substantially smaller increments in perplexity, falling within the **+0.546 to +0.810** range, and frequently results in improved or minimally altered MMLU scores, with gains up to +0.035.

D.2 Mean Replacement

Another approach of activation replacement discussed in the literature[Suau et al., 2021] is replacing it with the mean activation value. We provide the results for this type of replacement in Table 6.

The mean replacement strategy corresponds to setting the intervention function to $\phi(x) = \mu$, where μ is the mean activation of the neuron x computed over a general next-token prediction task on the Wikipedia[Foundation].

Table 6: Evaluation of Llama-3.2-3B on a DBPedia-14 dataset using neuron and range masking techniques. 30% neurons were selected. Mean Activation μ is used as replacement value. **Acc** represents class accuracy, **Conf** denotes class prediction probability, and **CAcc** and **CConf** refer to average accuracy and average class prediction probability across other classes, respectively. The *Base Values* indicate the baseline model performance, while *Neuron Masking* and *Activation Range Masking* show deviations from the baseline performance. PPL Δ and MMLU Δ show changes in perplexity and MMLU scores, respectively.

Class	Base Values				Neuron Masking						Activation Range Masking					
	Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf	PPL Δ	MMLU Δ	Acc	Conf	CAcc	CConf	PPL Δ	MMLU Δ
Class 0	1.000	0.576	1.000	0.563	-1.000	-0.576	-0.685	-0.554	7.681	-0.025	-1.000	-0.576	**-0.551**	**-0.545**	**0.687**	**-0.005**
Class 1	1.000	0.526	1.000	0.567	-1.000	-0.526	-0.554	-0.550	8.437	-0.030	-1.000	-0.526	**-0.356**	**-0.517**	**0.583**	**0.015**
Class 2	1.000	0.441	1.000	0.575	-0.995	-0.441	-0.697	-0.556	7.567	-0.015	-0.995	-0.440	**-0.574**	**-0.536**	**0.520**	**-0.010**
Class 3	1.000	0.461	1.000	0.573	-1.000	-0.461	-0.766	-0.561	8.005	-0.015	-1.000	-0.461	**-0.538**	**-0.534**	**0.543**	**0.010**
Class 4	1.000	0.839	1.000	0.541	-1.000	-0.838	-0.724	-0.528	8.239	**0.010**	-0.995	-0.838	**-0.502**	**-0.503**	**0.565**	0.005
Class 5	1.000	0.339	1.000	0.568	-1.000	-0.339	-0.616	-0.551	7.753	**0.010**	-1.000	-0.339	**-0.382**	**-0.510**	**0.552**	0.005
Class 6	1.000	0.810	1.000	0.545	-0.313	-0.805	-0.549	-0.531	7.880	-0.005	-0.292	-0.805	**-0.336**	**-0.499**	**0.547**	**0.020**
Class 7	1.000	0.595	1.000	0.562	-1.000	-0.592	-0.491	-0.535	7.413	-0.010	-0.995	-0.591	**-0.267**	**-0.449**	**0.462**	**0.000**
Class 8	1.000	0.417	1.000	0.574	-0.928	-0.414	-0.632	-0.556	7.688	0.015	-0.934	-0.414	**-0.298**	**-0.489**	**0.495**	0.015
Class 9	1.000	0.526	1.000	0.567	-1.000	-0.526	-0.611	-0.544	8.057	-0.035	-1.000	-0.526	**-0.370**	**-0.482**	**0.467**	**0.015**
Class 10	1.000	0.505	1.000	0.569	-0.998	-0.505	-0.642	-0.558	8.791	-0.020	-0.998	-0.505	**-0.406**	**-0.485**	**0.484**	**0.005**
Class 11	1.000	0.497	1.000	0.569	-1.000	-0.497	-0.719	-0.543	7.903	**0.025**	-1.000	-0.497	**-0.447**	**-0.459**	**0.397**	-0.005
Class 12	1.000	0.573	1.000	0.563	-0.904	-0.572	-0.629	-0.543	8.046	-0.005	-0.896	-0.571	**-0.375**	**-0.484**	**0.425**	**0.000**
Class 13	1.000	0.567	1.000	0.564	-1.000	-0.566	-0.526	-0.533	7.543	-0.025	-0.998	-0.566	**-0.341**	**-0.481**	**0.464**	**-0.010**

In Table 6, we assess the effect of mean replacement using both neuron masking and activation range masking. In every class, neuron masking results in more severe degradation than range-based masking across all auxiliary and general metrics.

Across metrics (Acc, Conf, CAcc, and CConf), activation range masking consistently outperforms neuron masking. The degradation in accuracy and confidence of auxiliary concepts is significantly lower under range-based interventions. For instance, in Class 3, neuron masking causes a drop of -1.000 in Acc and -0.766 in CAcc, whereas activation range masking yields a similar Acc drop (-1.000) but a substantially smaller decline in CAcc (-0.538). A similar pattern repeats across all classes; for example, in Class 0, neuron masking results in CAcc of -0.685 while activation range masking yields -0.551. In Class 7, neuron masking shows a CAcc of -0.491 compared to -0.267 for activation range masking.

Beyond auxiliary class performance, we observe substantial differences in how the two masking methods affect general language modelling capabilities. Neuron masking leads to a large increase in perplexity (PPL), ranging from **+7.413 to +8.791** which is catastrophic, across classes, and induces more negative shifts in MMLU scores (as low as -0.035 for Class 9, and also for Class 0 with -0.025, Class 1 with -0.030, Class 10 with -0.020, and Class 13 with -0.025). In contrast, activation range masking results in substantially smaller increases in perplexity (+0.397 to +0.687) and often yields improved or near-zero changes in MMLU scores (up to +0.020 for Class 6, and several positive values like +0.015 for Class 1, Class 8, and Class 9).

D.3 Adaptive Dampening

We propose a novel replacement method in which the intervention function $\phi(x)$ applies *runtime-controlled dampening* based on the distance of the observed activation x from the center of a predefined activation range. Specifically, the dampening factor $a(x)$ is linearly scaled according to the distance of x from the mean μ of the neuron's activation distribution, within the range $[\mu - 2.5\sigma, \mu + 2.5\sigma]$.

Let $\beta \in [0, 1]$ denote the maximum dampening factor applied at the range boundaries. Then:

$$a(x) = \beta \cdot \frac{|x - \mu|}{2.5\sigma}, \quad \text{and} \quad \phi(x) = a(x) \cdot x.$$

This ensures that when $x = \mu$ (the center of the range), $a(x) = 0$ and the activation is fully suppressed via $\phi(x) = 0$. At the boundaries ($x = \mu \pm 2.5\sigma$), $a(x) = \beta$, and the activation is minimally dampened. Values within the range are scaled proportionally based on their normalized distance from the mean. This adaptive dampening mechanism suppresses values near the mean while preserving those closer to the range edges.

The dampening factor β can be optimized for different neurons based on the concept information that neuron provides. For this work, we use $\beta = 0.5$ across all neurons.

Table 7: Evaluation of Llama-3.2-3B on a DBPedia-14 dataset using neuron and range masking techniques. 30% neurons were selected. Adaptive Dampening factor used is $\beta = 0.5$. **Acc** represents class accuracy, **Conf** denotes class prediction probability, and **CAcc** and **CConf** refer to average accuracy and average class prediction probability across other classes, respectively. The *Base Values* indicate the baseline model performance, while *Neuron Masking* and *Activation Range Masking* show deviations from the baseline performance. PPL Δ and MMLU Δ show changes in perplexity and MMLU scores, respectively.

Class	Base Values				Activation Range Masking					
	Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf	PPL Δ	MMLU Δ
Class 0	1.000	0.576	1.000	0.563	-0.927	-0.543	-0.215	-0.217	0.487	-0.015
Class 1	1.000	0.526	1.000	0.567	-0.791	-0.451	-0.134	-0.109	0.543	0.000
Class 2	1.000	0.441	1.000	0.575	-0.828	-0.380	-0.277	-0.215	0.540	-0.010
Class 3	1.000	0.461	1.000	0.573	-0.958	-0.432	-0.230	-0.214	0.492	0.010
Class 4	1.000	0.839	1.000	0.541	-0.346	-0.579	-0.261	-0.218	0.521	0.015
Class 5	1.000	0.339	1.000	0.568	-0.960	-0.319	-0.140	-0.116	0.609	-0.015
Class 6	1.000	0.810	1.000	0.545	-0.236	-0.613	-0.130	-0.122	0.524	-0.010
Class 7	1.000	0.595	1.000	0.562	-0.243	-0.388	-0.108	-0.080	0.408	0.005
Class 8	1.000	0.417	1.000	0.574	-0.440	-0.414	-0.152	-0.088	0.465	0.030
Class 9	1.000	0.526	1.000	0.567	-0.799	-0.459	-0.182	-0.131	0.445	0.005
Class 10	1.000	0.505	1.000	0.569	-0.684	-0.451	-0.222	-0.130	0.513	-0.010
Class 11	1.000	0.497	1.000	0.569	-0.836	-0.420	-0.308	-0.155	0.440	-0.005
Class 12	1.000	0.573	1.000	0.563	-0.720	-0.451	-0.172	-0.095	0.444	0.025
Class 13	1.000	0.567	1.000	0.564	-0.941	-0.530	-0.142	-0.098	0.502	0.010

In Table 7 we evaluate the adaptive dampening variant of the replacement function. This approach outperforms both neuron masking and static activation masking across all metrics.

In auxiliary class metrics, adaptive dampening yields much smaller degradation. Auxilary class accuracy (CAcc) and confidence (CConf) show significantly reduced drops compared to other methods. For example, in Class 0, CAcc drops only -0.215 compared to -0.685 under neuron masking and -0.551 under hard activation masking. The effect is consistent across classes, with most CAcc and CConf drops staying well below -0.3, and in many cases below -0.15.

Language modeling metrics show this approach to be exceptionally efficient. Perplexity increases are minimal, remaining within $+0.408$ to $+0.609$, substantially lower than all hard-masking variants. MMLU deltas also stay close to zero, with several classes showing improvement (e.g., Class 8: $+0.030$, Class 4: $+0.015$). Notably, no class suffers significant MMLU degradation.

E Training Details

For BERT, DistilBERT, and Llama, we utilize pretrained models. Since BERT, and DistilBert are not inherently trained as a conversational agent, we use top-performing fine-tuned models from the Hugging Face repository. For the Llama model, few-shot prompt completion is employed to predict class labels. This involves providing a small number of training samples from the dataset to guide the model's predictions.

For GPT-2, we fine-tune the pretrained model across all datasets for three epochs. The input sequence is constructed by concatenating the text with a <sep> token, followed by the class label, and ending with an <eos> token. During training, the loss is back-propagated only for the class label token, while all other tokens are assigned a skip label (-100). Additionally, all class labels are added to the model's dictionary as special single-token entries.

In the case of Bert-based models, record the activation of the CLS token, In the case of GPT-2 and Llama models, we record the last token output when the class token is being predicted. The intervention is applied to the appropriate token on the residual stream.

Dataset Preprocessing for Llama For Llama we process whole datasets in few shout settings and only curate 2000 samples per class, where the model prediction was correct.

F Compute Details

All experiments, including activation extraction and interventions on large language models (LLMs), were conducted using an NVIDIA RTX 3090 GPU equipped with 24GB of VRAM. 64GB RAM.

G Saliency details

Max activations. Frankle and Carbin [2019] extract high neural activations as a saliency ranking metric relying upon the rationale that maximally activating neurons are salient as these neurons play a critical role in controlling the model's output, highlighting their importance for a concept c.To identify them, the column-wise mean of absolute neuronal activations in H_c^l, H_c^l is defined in Section2.3, is computed, given that high negative activations also carry significant signals [Voita et al., 2023]. The magnitude of the means is then considered as a ranking for concept c.

Probe analysis. Dalvi et al. [2019b] train a linear classifier on the hidden representations H_c^l to distinguish between concepts. The learned model weights are then utilized as a saliency ranking. This process involves learning a weight matrix $W \in \mathbb{R}^{d \times |c|}$, where d is the hidden dimension and $|c|$ is the number of concept classes. The absolute weight values of each row in the weight matrix are used as a ranking for the importance of each neuron for a given concept. To prevent the emergence of redundant solutions characterized by minimal variations in the weights, the probe is trained using the elastic regularization technique.

Probeless. Antverg and Belinkov [2022] examine individual neurons, without the need for auxiliary classifiers, using the element-wise difference between mean vectors. The element-wise difference between mean vectors is computed as $r = \sum_{c,c' \in C} |q(c) - q(c')|$, where $r \in \mathbb{R}^d$ and d is the hidden dimension. The final neuron saliency ranking is obtained by sorting r in descending order.

Table 8: Performance drops relative to Baseline configuration (i.e.: unaltered model's performance) for three techniques: Probeless, Probe, and Max. All values show the difference from Base values. Results are for *Emotions* dataset on the GPT-2 model using 30% salient neurons of each method. Metrics are detailed in 2.1.

	Probeless				Probe				Max			
Class	**Acc**	**Conf**	**CAcc**	**CConf**	**Acc**	**Conf**	**CAcc**	**CConf**	**Acc**	**Conf**	**CAcc**	**CConf**
Class 0	**-0.738**	-0.733	-0.103	-0.097	-0.613	-0.650	**-0.010**	**-0.038**	-0.695	**-0.751**	-0.125	-0.124
Class 1	0.045	0.041	-0.113	-0.112	-0.014	-0.015	**-0.010**	-0.034	**-0.879**	**-0.882**	-0.019	**-0.009**
Class 2	-0.570	-0.541	-0.052	-0.057	0.017	0.009	-0.347	-0.359	**-0.776**	**-0.736**	**-0.029**	**-0.032**
Class 3	-0.164	-0.166	-0.035	-0.038	0.078	0.061	-0.047	-0.104	**-0.713**	**-0.714**	**-0.006**	**-0.007**
Class 4	-0.623	-0.617	-0.087	-0.084	-0.005	-0.010	**-0.003**	**-0.020**	**-0.754**	**-0.753**	-0.240	-0.248
Class 5	**-0.817**	**-0.714**	-0.101	-0.105	-0.206	-0.127	**0.003**	**-0.010**	-0.587	-0.601	-0.301	-0.308

H Hyperparameter Ablation

For target concept, τ values $0.3 - 2.4$ show decreasing accuracy/confidence, stabilizing at $\tau = 2.4$ (accuracy 0.6126). Beyond 2.4, negligible additional degradation occurs, indicating we've captured the complete target concept activation range. Importantly, while target performance stabilizes after $\tau = 2.4$, auxiliary task performance declines after $\tau = 2.7$. Complement accuracy stays above 0.93 until then before dropping to 0.8795 at $\tau = 4.5$. This aligns with normal distribution properties where 95-99% of values fall within ±2.5 standard deviations.

Table 9: Performance metrics for varying τ values.

τ	Acc	Conf	CAcc	CConf
0.3	0.9021	0.8858	0.9452	0.9358
0.6	0.8439	0.8185	0.9424	0.9327
0.9	0.7801	0.7486	0.9391	0.9263
1.2	0.7295	0.6950	0.9340	0.9174
1.5	0.6834	0.6482	0.9337	0.9093
1.8	0.6424	0.6141	0.9331	0.9000
2.1	0.6184	0.5926	0.9327	0.8910
2.4	0.6126	0.5858	0.9314	0.8846
2.7	0.6024	0.5798	0.9280	0.8800
3.0	0.5971	0.5776	0.9234	0.8777
3.3	0.5963	0.5786	0.9173	0.8753
3.6	0.5970	0.5794	0.9097	0.8729
3.9	0.5976	0.5802	0.9020	0.8698
4.2	0.5967	0.5798	0.8908	0.8642
4.5	0.5967	0.5798	0.8795	0.8577

I Layer Ablation

I.1 Statistical Results

We analyze concept level activation distributions across all 12 layers of GPT-2, measuring kurtosis (where a value of 3.0 indicates a Gaussian distribution), skewness (where 0 indicates symmetry), and practical normality in Table 10:

Table 10: Statistical analysis of different layers showing skewness, kurtosis, and Kolmogorov-Smirnov test results. *GPT2* model. *AG-News Dataset*

Layer	Kurtosis	Skewness	Practical Normality(10%)
1	3.9314	0.0430	0.7913
2	3.7622	-0.0091	0.9525
3	3.4109	-0.0143	0.9870
4	3.5582	-0.0073	0.9801
5	3.6145	0.0051	0.9730
6	3.5318	0.0086	0.9769
7	3.3461	0.0083	0.9880
8	3.2763	0.0037	0.9870
9	3.2267	0.0039	0.9860
10	3.2057	0.0029	0.9899
11	3.2105	-0.0002	0.9912
12	3.2061	-0.0014	0.9919

These results show that kurtosis values converge toward 3.0 (the Gaussian ideal) as layers progress, skewness values remain near zero across all layers, and practical normality scores are close to 1 across all layers. Importantly, if the activations were not clustered into continuous intervals and were in disconnected islands of activations, these would be reflected in the score for the practical normality and other statistical metrics.

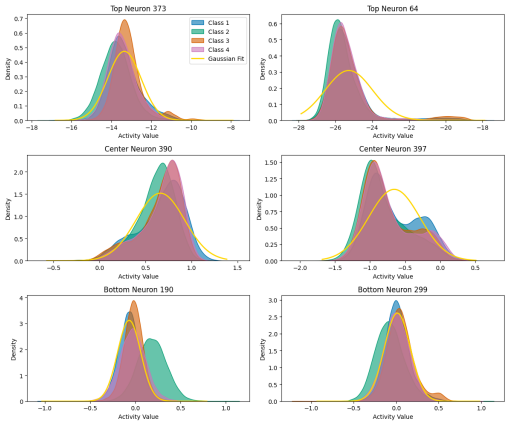

Figure 7: Neuronal Activation Patterns of six neurons on *AG-News* dataset. Layer 1

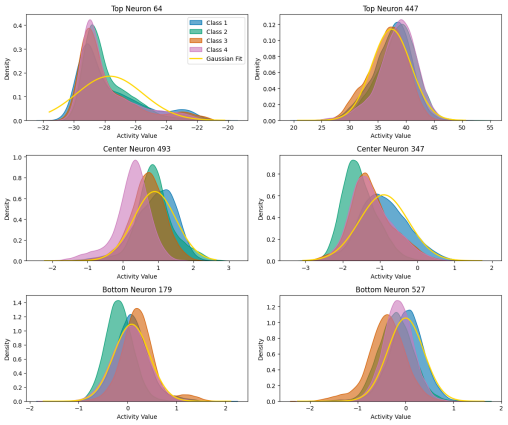

Figure 8: Neuronal Activation Patterns of six neurons on *AG-News* dataset. Layer 2

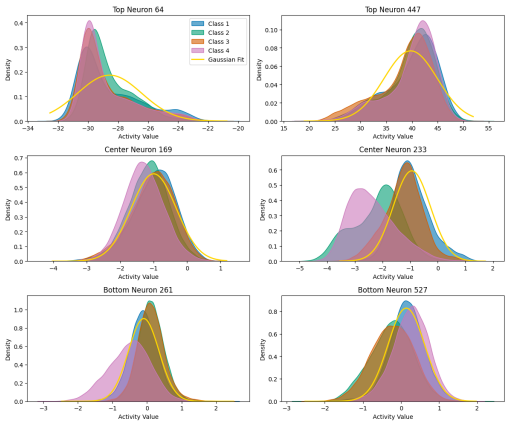

Figure 9: Neuronal Activation Patterns of six neurons on *AG-News* dataset. Layer 3

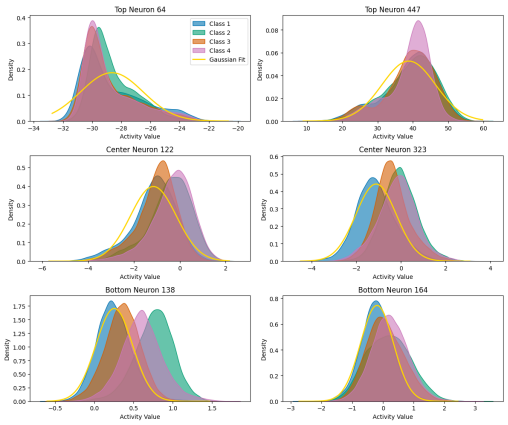

Figure 10: Neuronal Activation Patterns of six neurons on *AG-News* dataset. Layer 4

I.2 Qualitative Results

We expanded our visualization approach shown in Figures figs. 7 to 18 in Figure 4) to all layers in the model. The visualizations demonstrate an interesting progression: while all layers exhibit Gaussian-like distributions on the class level, class concepts aren't separated in the activation spectrum Gaussians of the early layers. This aligns with the understanding that lower layers capture more basic features rather than high-level semantic features like class. However, distinct concept-level Gaussian distributions begin forming as early as layers 5-6, becoming increasingly separable in deeper layers.

I.3 Masking Results

In Table 11 and Table 12 we provide results of applying both approaches on all layers of *GPT-2* model on *Emotions* dataset. From the results we can see that: Early layers (1-3) show highly variable and often severe impacts: Layer 1 exhibits minimal effects ($\Delta Acc = -0.113$, $\Delta CAcc = -0.064$), while Layers 2-3 show extreme degradation ($\Delta Acc \approx -0.7$, $\Delta CAcc > -0.5$). Middle layers (4-8) demonstrate inconsistent behavior with high variance in impacts. Layer 12, however, achieves an optimal balance: it maintains substantial primary task impact ($\Delta Acc = -0.571$) while minimizing auxiliary concept interference ($\Delta CAcc = -0.060$). This pattern holds true for both neuron masking and range masking techniques, with range masking showing slightly better preservation of auxiliary concepts ($\Delta CAcc = -0.045$). The mid-range primary task degradation combined with minimal auxiliary impact makes Layer 12 the most suitable for targeted interventions, offering better control and specificity compared to earlier layers.

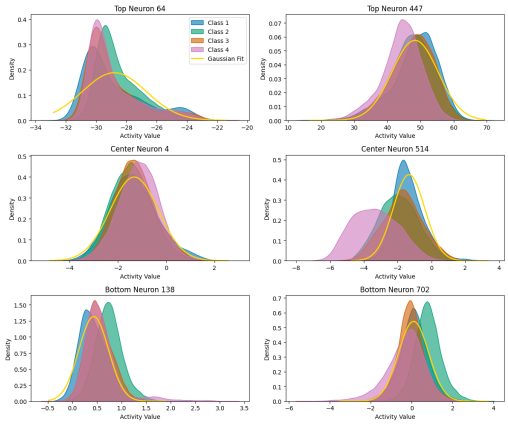

Figure 11: Neuronal Activation Patterns of six neurons on *AG-News* dataset. Layer 5

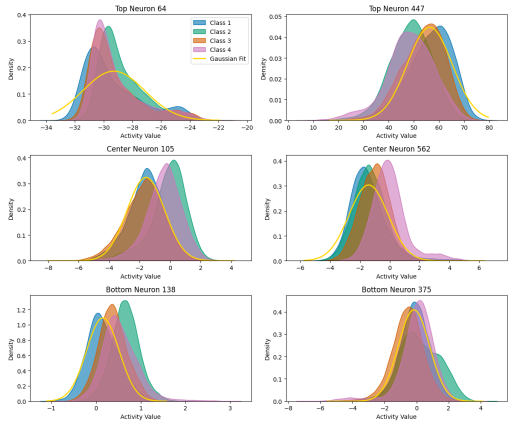

Figure 12: Neuronal Activation Patterns of six neurons on *AG-News* dataset. Layer 6

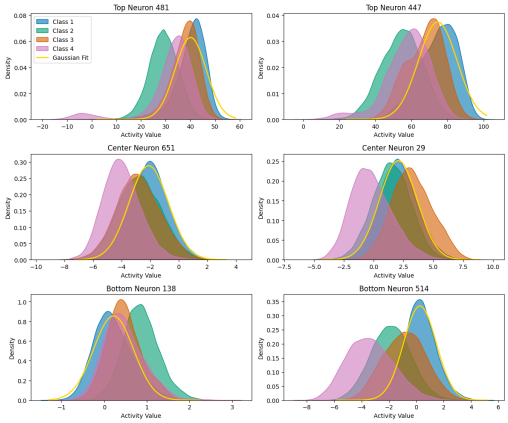

Figure 13: Neuronal Activation Patterns of six neurons on *AG-News* dataset. Layer 7

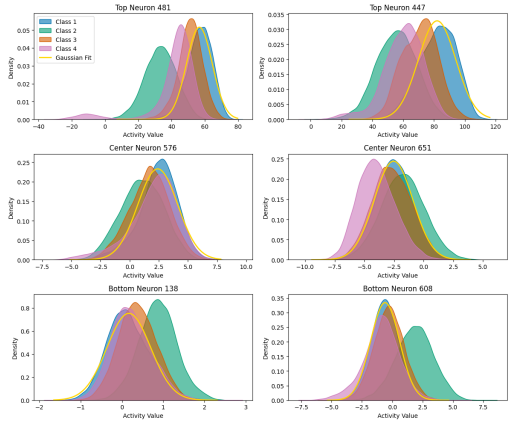

Figure 14: Neuronal Activation Patterns of six neurons on *AG-News* dataset. Layer 8

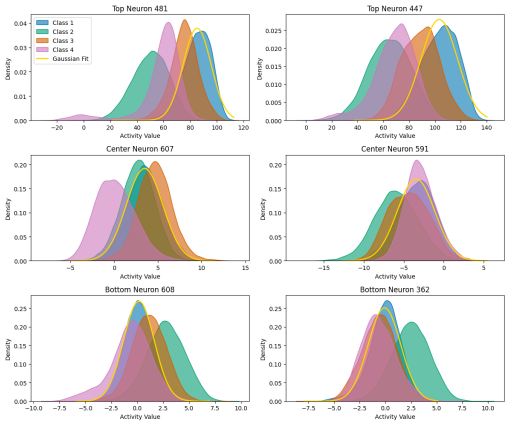

Figure 15: Neuronal Activation Patterns of six neurons on *AG-News* dataset. Layer 9

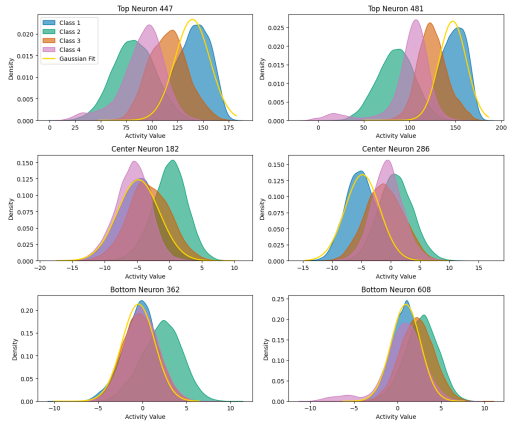

Figure 16: Neuronal Activation Patterns of six neurons on *AG-News* dataset. Layer 10

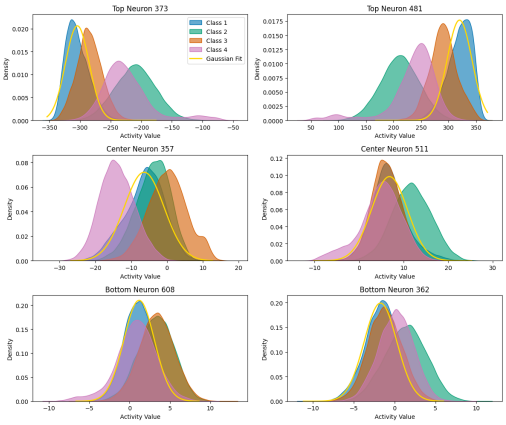

Figure 17: Neuronal Activation Patterns of six neurons on *AG-News* dataset. Layer 11

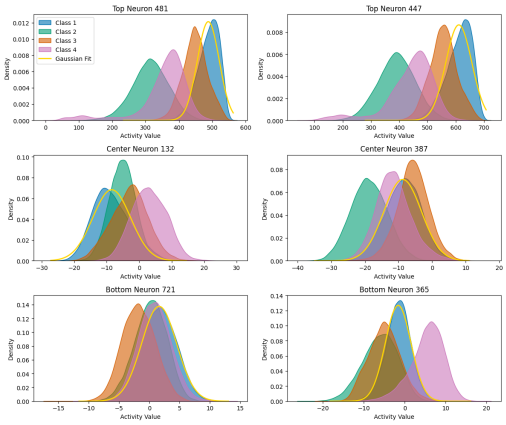

Figure 18: Neuronal Activation Patterns of six neurons on *AG-News* dataset. Layer 12

J Class Wise Results

Here we provide the complete results for the datasets shown in Table 3. In Table 14 we provide results on *IMDB* dataset on all selected models. In Table 15 we provide results on *SST2* dataset on all selected models. In Table 16 we provide results on *Emotions* dataset on all selected models. In Table 17 we provide results on *DBPedia-14* dataset on all selected models.

Table 11: Evaluation of layer selection on *GPT-2* model on the *Emotions* dataset using neuron and range masking techniques. 20% Neurons selected. Here, **Acc** represents class accuracy, **Conf** denotes class prediction probability, and **CAcc** and **CConf** refer to average accuracy and average class prediction probability across other classes, respectively. The *Base Values* indicate the baseline model performance, while *Activation Range Masking* and *Neuron Masking* show deviations from the baseline performance.

Layer	Class	Base Values				Neuron Masking				Activation Range Masking			
		Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf
1	Class 0	0.970	0.957	0.915	0.904	-0.029	-0.071	-0.074	-0.100	0.006	0.002	-0.004	-0.005
	Class 1	0.933	0.932	0.931	0.913	-0.011	-0.056	-0.090	-0.116	0.001	-0.003	-0.004	-0.004
	Class 2	0.901	0.865	0.934	0.924	-0.206	-0.195	-0.052	-0.092	-0.019	-0.015	-0.001	-0.002
	Class 3	0.926	0.924	0.932	0.919	-0.128	-0.152	-0.051	-0.090	-0.004	-0.005	-0.001	-0.002
	Class 4	0.885	0.867	0.938	0.927	-0.055	-0.084	-0.061	-0.093	-0.016	-0.009	0.002	-0.001
	Class 5	0.851	0.786	0.934	0.924	-0.249	-0.217	-0.055	-0.094	0.016	0.013	-0.004	-0.005
2	Class 0	0.970	0.957	0.915	0.904	-0.804	-0.808	-0.389	-0.386	-0.061	-0.133	-0.077	-0.096
	Class 1	0.933	0.932	0.931	0.913	0.053	-0.003	-0.819	-0.781	-0.011	-0.049	-0.110	-0.145
	Class 2	0.901	0.865	0.934	0.924	-0.868	-0.737	-0.515	-0.519	-0.365	-0.337	-0.077	-0.126
	Class 3	0.926	0.924	0.932	0.919	-0.870	-0.805	-0.498	-0.501	-0.215	-0.248	-0.096	-0.153
	Class 4	0.885	0.867	0.938	0.927	-0.729	-0.707	-0.461	-0.463	-0.042	-0.077	-0.076	-0.116
	Class 5	0.851	0.786	0.934	0.924	-0.845	-0.769	-0.511	-0.508	-0.229	-0.188	-0.106	-0.163
3	Class 0	0.970	0.957	0.915	0.904	-0.896	-0.904	-0.824	-0.832	-0.647	-0.688	-0.517	-0.544
	Class 1	0.933	0.932	0.931	0.913	-0.901	-0.916	-0.835	-0.832	-0.568	-0.607	-0.609	-0.630
	Class 2	0.901	0.865	0.934	0.924	-0.868	-0.845	-0.838	-0.851	-0.605	-0.600	-0.589	-0.619
	Class 3	0.926	0.924	0.932	0.919	-0.868	-0.896	-0.830	-0.840	-0.567	-0.605	-0.567	-0.596
	Class 4	0.885	0.867	0.938	0.927	-0.800	-0.811	-0.849	-0.857	-0.502	-0.522	-0.513	-0.544
	Class 5	0.851	0.786	0.934	0.924	0.022	0.081	-0.865	-0.881	-0.155	-0.124	-0.561	-0.596
4	Class 0	0.970	0.957	0.915	0.904	-0.650	-0.703	-0.698	-0.764	-0.608	-0.621	-0.499	-0.510
	Class 1	0.933	0.932	0.931	0.913	-0.845	-0.884	-0.667	-0.725	-0.491	-0.519	-0.480	-0.491
	Class 2	0.901	0.865	0.934	0.924	-0.858	-0.824	-0.772	-0.809	-0.488	-0.497	-0.506	-0.523
	Class 3	0.926	0.924	0.932	0.919	-0.700	-0.808	-0.663	-0.739	-0.534	-0.546	-0.512	-0.528
	Class 4	0.885	0.867	0.938	0.927	-0.239	-0.514	-0.754	-0.797	-0.304	-0.307	-0.452	-0.471
	Class 5	0.851	0.786	0.934	0.924	-0.612	-0.463	-0.692	-0.765	-0.047	-0.038	-0.525	-0.541
5	Class 0	0.970	0.957	0.915	0.904	-0.838	-0.852	-0.492	-0.630	-0.695	-0.688	-0.554	-0.555
	Class 1	0.933	0.932	0.931	0.913	-0.387	-0.563	-0.683	-0.714	-0.552	-0.564	-0.605	-0.599
	Class 2	0.901	0.865	0.934	0.924	-0.702	-0.700	-0.634	-0.690	-0.472	-0.470	-0.607	-0.605
	Class 3	0.926	0.924	0.932	0.919	-0.361	-0.507	-0.615	-0.692	-0.567	-0.575	-0.538	-0.539
	Class 4	0.885	0.867	0.938	0.927	-0.873	-0.844	-0.525	-0.650	-0.668	-0.653	-0.594	-0.594
	Class 5	0.851	0.786	0.934	0.924	-0.637	-0.573	-0.588	-0.681	-0.069	-0.022	-0.548	-0.553
6	Class 0	0.970	0.957	0.915	0.904	-0.720	-0.775	-0.829	-0.830	-0.484	-0.499	-0.318	-0.322
	Class 1	0.933	0.932	0.931	0.913	-0.871	-0.887	-0.750	-0.768	-0.176	-0.195	-0.499	-0.499
	Class 2	0.901	0.865	0.934	0.924	-0.895	-0.860	-0.735	-0.773	-0.680	-0.638	-0.335	-0.348
	Class 3	0.926	0.924	0.932	0.919	-0.863	-0.884	-0.772	-0.793	-0.418	-0.431	-0.379	-0.381
	Class 4	0.885	0.867	0.938	0.927	-0.621	-0.669	-0.743	-0.784	-0.430	-0.435	-0.247	-0.262
	Class 5	0.851	0.786	0.934	0.924	-0.143	-0.086	-0.808	-0.831	-0.114	-0.070	-0.474	-0.478

Table 12: Evaluation of layer selection on *GPT-2* model on the *Emotions* dataset using neuron and range masking techniques. 20% Neurons selected. Here, **Acc** represents class accuracy, **Conf** denotes class prediction probability, and **CAcc** and **CConf** refer to average accuracy and average class prediction probability across other classes, respectively. The *Base Values* indicate the baseline model performance, while *Activation Range Masking* and *Neuron Masking* show deviations from the baseline performance.

Layer	Class	Base Values				Neuron Masking				Activation Range Masking			
		Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf
7	Class 0	0.970	0.957	0.915	0.904	-0.908	-0.901	-0.752	-0.753	-0.527	-0.538	-0.492	-0.498
	Class 1	0.933	0.932	0.931	0.913	-0.884	-0.895	-0.743	-0.729	-0.484	-0.509	-0.330	-0.338
	Class 2	0.901	0.865	0.934	0.924	-0.866	-0.835	-0.767	-0.765	-0.451	-0.442	-0.336	-0.355
	Class 3	0.926	0.924	0.932	0.919	-0.786	-0.819	-0.641	-0.666	-0.445	-0.457	-0.331	-0.346
	Class 4	0.885	0.867	0.938	0.927	-0.626	-0.618	-0.810	-0.817	-0.341	-0.335	-0.521	-0.532
	Class 5	0.851	0.786	0.934	0.924	0.106	0.147	-0.810	-0.811	0.102	0.107	-0.547	-0.553
8	Class 0	0.970	0.957	0.915	0.904	-0.776	-0.791	-0.209	-0.291	-0.191	-0.312	-0.082	-0.114
	Class 1	0.933	0.932	0.931	0.913	-0.585	-0.667	-0.412	-0.441	-0.591	-0.644	-0.199	-0.227
	Class 2	0.901	0.865	0.934	0.924	-0.692	-0.716	-0.469	-0.496	-0.560	-0.562	-0.468	-0.486
	Class 3	0.926	0.924	0.932	0.919	-0.657	-0.714	-0.415	-0.464	-0.468	-0.503	-0.230	-0.266
	Class 4	0.885	0.867	0.938	0.927	-0.501	-0.509	-0.531	-0.569	-0.201	-0.234	-0.258	-0.290
	Class 5	0.851	0.786	0.934	0.924	-0.092	-0.050	-0.634	-0.647	0.065	0.058	-0.279	-0.308
9	Class 0	0.970	0.957	0.915	0.904	-0.759	-0.768	-0.311	-0.351	-0.610	-0.661	-0.307	-0.328
	Class 1	0.933	0.932	0.931	0.913	-0.570	-0.713	-0.319	-0.346	-0.906	-0.910	-0.267	-0.298
	Class 2	0.901	0.865	0.934	0.924	-0.424	-0.520	-0.504	-0.531	-0.635	-0.643	-0.579	-0.595
	Class 3	0.926	0.924	0.932	0.919	-0.810	-0.834	-0.501	-0.502	-0.759	-0.772	-0.502	-0.516
	Class 4	0.885	0.867	0.938	0.927	-0.358	-0.357	-0.476	-0.481	-0.587	-0.566	-0.519	-0.527
	Class 5	0.851	0.786	0.934	0.924	-0.133	-0.101	-0.546	-0.554	0.106	0.104	-0.450	-0.462
10	Class 0	0.970	0.957	0.915	0.904	-0.733	-0.741	-0.105	-0.126	-0.624	-0.659	-0.146	-0.163
	Class 1	0.933	0.932	0.931	0.913	-0.389	-0.671	-0.178	-0.209	-0.899	-0.911	-0.254	-0.285
	Class 2	0.901	0.865	0.934	0.924	-0.230	-0.513	-0.116	-0.224	-0.699	-0.735	-0.409	-0.451
	Class 3	0.926	0.924	0.932	0.919	-0.434	-0.687	-0.081	-0.133	-0.898	-0.905	-0.401	-0.455
	Class 4	0.885	0.867	0.938	0.927	-0.489	-0.506	-0.188	-0.256	-0.140	-0.186	-0.063	-0.102
	Class 5	0.851	0.786	0.934	0.924	-0.306	-0.243	-0.157	-0.240	0.063	0.010	-0.095	-0.127
11	Class 0	0.970	0.957	0.915	0.904	-0.358	-0.496	-0.382	-0.414	-0.301	-0.441	-0.121	-0.148
	Class 1	0.933	0.932	0.931	0.913	-0.800	-0.857	-0.078	-0.123	-0.858	-0.875	-0.128	-0.162
	Class 2	0.901	0.865	0.934	0.924	-0.897	-0.861	-0.416	-0.450	-0.901	-0.864	-0.464	-0.500
	Class 3	0.926	0.924	0.932	0.919	-0.923	-0.921	-0.427	-0.470	-0.913	-0.914	-0.354	-0.393
	Class 4	0.885	0.867	0.938	0.927	-0.152	-0.212	-0.039	-0.075	-0.210	-0.239	-0.181	-0.204
	Class 5	0.851	0.786	0.934	0.924	0.047	-0.028	-0.131	-0.173	0.053	0.002	-0.142	-0.159
12	Class 0	0.970	0.957	0.915	0.904	-0.550	-0.603	-0.013	-0.003	-0.542	-0.594	0.005	0.012
	Class 1	0.933	0.932	0.931	0.913	-0.526	-0.545	0.001	0.012	-0.521	-0.538	-0.005	-0.004
	Class 2	0.901	0.865	0.934	0.924	-0.416	-0.402	0.002	0.006	-0.419	-0.407	0.007	0.006
	Class 3	0.926	0.924	0.932	0.919	-0.561	-0.576	-0.007	0.003	-0.561	-0.572	0.000	0.005
	Class 4	0.885	0.867	0.938	0.927	-0.655	-0.658	-0.042	-0.034	-0.657	-0.659	-0.011	-0.003
	Class 5	0.851	0.786	0.934	0.924	-0.718	-0.672	-0.300	-0.297	-0.718	-0.672	-0.267	-0.266

Table 13: Evaluation of selected models on the *AG-News* dataset using neuron and range masking techniques. **Acc** represents class accuracy, **Conf** denotes class prediction probability, and **CAcc** and **CConf** refer to average accuracy and average class prediction probability across other classes, respectively. The *Base Values* indicate the baseline model performance, while *Activation Range Masking* and *Neuron Masking* show deviations from the baseline performance. For *GPT-2* 50% and for *Llama-3.2-3B* 30% neurons selected.

Model	Class	Base Values				Neuron Masking				Activation Range Masking			
		Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf
BERT	Class 0	0.945	0.936	0.949	0.927	-0.205	-0.587	**0.004**	-0.076	-0.198	-0.589	0.007	**-0.010**
	Class 1	0.993	0.988	0.933	0.910	-0.225	-0.659	0.004	-0.077	-0.194	-0.650	**0.003**	**-0.012**
	Class 2	0.905	0.881	0.962	0.945	-0.300	-0.536	0.014	-0.079	-0.298	-0.542	0.014	**-0.009**
	Class 3	0.949	0.913	0.948	0.935	-0.354	-0.577	0.026	-0.065	-0.353	-0.579	**0.025**	**-0.005**
GPT-2	Class 0	0.955	0.951	0.941	0.928	-0.920	-0.926	-0.231	-0.224	-0.919	-0.925	**-0.019**	**-0.008**
	Class 1	0.986	0.981	0.931	0.917	-0.926	-0.931	-0.253	-0.257	-0.912	-0.916	**-0.054**	**-0.069**
	Class 2	0.897	0.886	0.960	0.949	-0.696	-0.737	-0.110	**-0.132**	-0.678	-0.725	**-0.097**	-0.306
	Class 3	0.940	0.916	0.946	0.939	-0.940	-0.916	-0.024	**-0.037**	-0.887	-0.882	**-0.080**	-0.510
Llama-3.2-3B	Class 0	1.000	0.936	1.000	0.680	-0.995	-0.934	-0.530	-0.427	-0.995	-0.934	**-0.345**	**-0.306**
	Class 1	1.000	0.742	1.000	0.744	-0.870	-0.680	-0.615	-0.599	-0.875	-0.681	**-0.515**	**-0.503**
	Class 2	1.000	0.655	1.000	0.773	-0.895	-0.646	-0.795	-0.634	-0.895	-0.646	**-0.655**	**-0.549**
	Class 3	1.000	0.642	1.000	0.778	-0.975	-0.641	-0.698	-0.630	-0.975	-0.640	**-0.420**	**-0.459**

Table 14: Evaluation of selected models on the *IMDB* dataset using neuron and range masking techniques. Here, **Acc** represents class accuracy, **Conf** denotes class prediction probability, and **CAcc** and **CConf** refer to average accuracy and average class prediction probability across other classes, respectively. The *Base Values* indicate the baseline model performance, while *Activation Range Masking* and *Neuron Masking* show deviations from the baseline performance.

Model	Class	Base Values				Neuron Masking				Activation Range Masking			
		Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf
BERT	Class 0	0.930	0.908	0.926	0.901	-0.169	-0.352	0.061	-0.066	-0.163	-0.359	0.059	0.035
	Class 1	0.926	0.901	0.930	0.908	-0.211	-0.355	0.057	-0.091	-0.206	-0.361	0.056	0.025
GPT-2	Class 0	0.965	0.941	0.940	0.922	-0.935	-0.922	0.050	0.057	-0.905	-0.901	0.055	0.046
	Class 1	0.940	0.922	0.965	0.941	-0.620	-0.667	0.005	0.018	-0.610	-0.657	0.015	0.027
Llama-3.2-3B	Class 0	1.000	0.619	1.000	0.500	-0.643	-0.448	-0.515	-0.287	-0.640	-0.446	-0.502	-0.278
	Class 1	1.000	0.500	1.000	0.619	-0.877	-0.410	-0.273	-0.304	-0.873	-0.409	-0.265	-0.303

Table 15: Evaluation of selected models on the *SST2* dataset using neuron and range masking techniques. Here, **Acc** represents class accuracy, **Conf** denotes class prediction probability, and **CAcc** and **CConf** refer to average accuracy and average class prediction probability across other classes, respectively. The *Base Values* indicate the baseline model performance, while *Activation Range Masking* and *Neuron Masking* show deviations from the baseline performance.

Model	Class	Base Values				Neuron Masking				Activation Range Masking			
		Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf
BERT	Class 0	0.890	0.882	0.930	0.925	-0.058	-0.308	0.029	-0.047	-0.075	-0.329	0.031	0.036
	Class 1	0.930	0.925	0.890	0.882	-0.043	-0.318	0.033	-0.045	-0.045	-0.330	0.030	0.050
GPT-2	Class 0	0.950	0.937	0.981	0.978	-0.142	-0.158	0.010	0.012	-0.142	-0.167	0.009	0.010
	Class 1	0.981	0.978	0.950	0.937	-0.187	-0.223	0.041	0.053	-0.176	-0.216	0.041	0.046
Llama-3.2-3B	Class 0	1.000	0.620	1.000	0.690	-0.532	-0.459	-0.420	-0.424	-0.532	-0.456	-0.404	-0.415
	Class 1	1.000	0.690	1.000	0.620	-0.289	-0.379	-0.326	-0.315	-0.284	-0.376	-0.306	-0.301

Table 16: Evaluation of selected models on the *Emotions* dataset using neuron and range masking techniques. Here, **Acc** represents class accuracy, **Conf** denotes class prediction probability, and **CAcc** and **CConf** refer to average accuracy and average class prediction probability across other classes, respectively. The *Base Values* indicate the baseline model performance, while *Activation Range Masking* and *Neuron Masking* show deviations from the baseline performance.

Model	Class	Base Values				Neuron Masking				Activation Range Masking			
		Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf
BERT	Class 0	0.960	0.935	0.901	0.851	-0.241	-0.718	0.013	-0.266	-0.222	-0.718	0.012	-0.055
	Class 1	0.942	0.904	0.905	0.861	-0.223	-0.691	0.028	-0.254	-0.213	-0.692	0.032	-0.064
	Class 2	0.824	0.723	0.926	0.889	-0.371	-0.533	0.016	-0.284	-0.352	-0.534	0.018	-0.115
	Class 3	0.927	0.873	0.916	0.876	-0.247	-0.664	0.010	-0.256	-0.240	-0.667	0.012	-0.057
	Class 4	0.884	0.837	0.922	0.880	-0.406	-0.646	0.012	-0.251	-0.402	-0.648	0.012	-0.066
	Class 5	0.591	0.566	0.929	0.886	-0.303	-0.392	0.004	-0.299	-0.303	-0.397	0.005	-0.090
GPT-2	Class 0	0.969	0.956	0.913	0.903	-0.695	-0.751	-0.125	-0.124	-0.698	-0.749	-0.009	-0.009
	Class 1	0.939	0.938	0.925	0.908	-0.879	-0.882	-0.019	-0.009	-0.879	-0.880	-0.016	-0.015
	Class 2	0.902	0.872	0.932	0.923	-0.776	-0.736	-0.029	-0.032	-0.780	-0.739	-0.023	-0.028
	Class 3	0.910	0.905	0.932	0.921	-0.713	-0.714	-0.006	-0.007	-0.715	-0.716	-0.002	-0.001
	Class 4	0.869	0.854	0.938	0.927	-0.754	-0.753	-0.240	-0.248	-0.754	-0.753	-0.127	-0.133
	Class 5	0.857	0.798	0.932	0.923	-0.587	-0.601	-0.301	-0.308	-0.587	-0.601	-0.280	-0.289
Llama-3.2-3B	Class 0	0.950	0.550	0.782	0.455	-0.950	-0.547	-0.655	-0.408	-0.945	-0.547	-0.571	-0.378
	Class 1	0.905	0.498	0.804	0.473	-0.855	-0.495	-0.743	-0.433	-0.867	-0.494	-0.607	-0.404
	Class 2	0.785	0.421	0.827	0.483	-0.785	-0.420	-0.771	-0.454	-0.785	-0.420	-0.658	-0.436
	Class 3	0.790	0.482	0.833	0.476	-0.760	-0.477	-0.635	-0.423	-0.755	-0.476	-0.544	-0.402
	Class 4	0.780	0.487	0.829	0.476	-0.780	-0.486	-0.534	-0.365	-0.780	-0.486	-0.444	-0.324
	Class 5	0.536	0.296	0.855	0.498	-0.417	-0.284	-0.751	-0.465	-0.429	-0.282	-0.653	-0.434

Table 17: Evaluation of selected models on the *DBPedia-14* dataset using neuron and range masking techniques. Here, **Acc** represents class accuracy, **Conf** denotes class prediction probability, and **CAcc** and **CConf** refer to average accuracy and average class prediction probability across other classes, respectively. The *Base Values* indicate the baseline model performance, while *Activation Range Masking* and *Neuron Masking* show deviations from the baseline performance.

Model	Class	Base Values				Neuron Masking				Activation Range Masking			
		Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf	Acc	Conf	CAcc	CConf
BERT	Class 0	0.972	0.966	0.992	0.991	-0.082	-0.702	0.001	-0.014	-0.076	-0.698	0.001	-0.000
	Class 1	0.987	0.986	0.991	0.990	-0.030	-0.778	0.000	-0.017	-0.018	-0.770	0.000	-0.000
	Class 2	0.987	0.985	0.991	0.990	-0.239	-0.814	0.001	-0.018	-0.217	-0.806	0.001	-0.000
	Class 3	0.997	0.997	0.990	0.989	-0.008	-0.766	0.000	-0.019	-0.001	-0.731	0.000	-0.000
	Class 4	0.984	0.983	0.991	0.990	-0.058	-0.777	0.001	-0.018	-0.032	-0.761	0.000	-0.000
	Class 5	0.995	0.995	0.990	0.989	-0.007	-0.795	0.000	-0.017	-0.001	-0.771	0.000	-0.000
	Class 6	0.975	0.974	0.992	0.991	-0.121	-0.807	0.000	-0.015	-0.112	-0.803	0.000	-0.001
	Class 7	0.994	0.994	0.990	0.989	-0.028	-0.789	0.000	-0.017	-0.010	-0.767	0.000	-0.000
	Class 8	1.000	1.000	0.990	0.989	-0.001	-0.808	0.000	-0.022	0.000	-0.772	0.000	-0.000
	Class 9	0.999	0.998	0.990	0.989	-0.004	-0.837	0.000	-0.019	-0.001	-0.811	0.000	-0.000
	Class 10	0.994	0.993	0.990	0.989	-0.025	-0.846	0.000	-0.016	-0.005	-0.831	0.000	-0.000
	Class 11	0.997	0.997	0.990	0.989	-0.013	-0.751	0.000	-0.017	-0.001	-0.726	0.000	-0.000
	Class 12	0.990	0.990	0.990	0.989	-0.018	-0.772	0.000	-0.017	-0.005	-0.755	0.000	-0.000
	Class 13	0.994	0.994	0.990	0.989	-0.009	-0.740	0.001	-0.017	-0.001	-0.721	0.000	-0.000
GPT-2	Class 0	0.985	0.977	0.990	0.989	-0.860	-0.877	-0.133	-0.136	-0.850	-0.869	-0.002	-0.017
	Class 1	0.995	0.992	0.990	0.988	-0.500	-0.567	-0.180	-0.192	-0.460	-0.544	-0.023	-0.024
	Class 2	0.985	0.980	0.990	0.989	-0.890	-0.904	-0.189	-0.213	-0.880	-0.902	-0.004	-0.010
	Class 3	0.995	0.995	0.990	0.987	-0.900	-0.933	-0.145	-0.143	-0.900	-0.927	-0.008	-0.017
	Class 4	0.970	0.969	0.992	0.989	-0.715	-0.773	-0.224	-0.260	-0.695	-0.750	-0.042	-0.062
	Class 5	0.995	0.993	0.990	0.988	-0.315	-0.446	-0.127	-0.192	-0.290	-0.432	-0.013	-0.025
	Class 6	0.965	0.964	0.992	0.990	-0.925	-0.932	-0.052	-0.062	-0.910	-0.928	-0.006	-0.007
	Class 7	1.000	0.998	0.989	0.987	-0.815	-0.865	-0.003	-0.008	-0.775	-0.846	-0.026	-0.057
	Class 8	1.000	1.000	0.989	0.987	-0.995	-0.990	-0.148	-0.188	-0.900	-0.932	-0.026	-0.055
	Class 9	1.000	1.000	0.989	0.987	-0.975	-0.979	-0.250	-0.268	-0.955	-0.958	-0.020	-0.049
	Class 10	0.995	0.993	0.990	0.988	-0.595	-0.685	-0.045	-0.053	-0.590	-0.675	-0.005	-0.011
	Class 11	0.985	0.984	0.990	0.988	-0.210	-0.453	-0.094	-0.118	-0.135	-0.396	-0.015	-0.034
	Class 12	0.990	0.988	0.990	0.988	-0.930	-0.938	-0.293	-0.309	-0.855	-0.880	-0.013	-0.029
	Class 13	1.000	0.999	0.989	0.987	-0.985	-0.986	-0.393	-0.416	-0.945	-0.981	-0.018	-0.044
Llama-3.2-3B	Class 0	1.000	0.586	1.000	0.559	-0.990	-0.584	-0.949	-0.473	-0.990	-0.584	-0.823	-0.441
	Class 1	1.000	0.533	1.000	0.563	-1.000	-0.528	-0.870	-0.446	-0.970	-0.528	-0.706	-0.371
	Class 2	1.000	0.467	1.000	0.568	-0.995	-0.462	-0.963	-0.477	-0.995	-0.461	-0.838	-0.432
	Class 3	1.000	0.460	1.000	0.569	-0.995	-0.459	-0.981	-0.486	-0.995	-0.459	-0.815	-0.420
	Class 4	1.000	0.828	1.000	0.539	-0.965	-0.809	-0.981	-0.454	-0.955	-0.808	-0.852	-0.412
	Class 5	1.000	0.349	1.000	0.568	-1.000	-0.348	-0.882	-0.429	-0.989	-0.347	-0.585	-0.346
	Class 6	1.000	0.809	1.000	0.541	-1.000	-0.787	-0.972	-0.449	-1.000	-0.787	-0.736	-0.366
	Class 7	1.000	0.599	1.000	0.558	-0.855	-0.588	-0.918	-0.410	-0.860	-0.586	-0.489	-0.274
	Class 8	1.000	0.420	1.000	0.572	-1.000	-0.420	-0.957	-0.467	-1.000	-0.420	-0.660	-0.335
	Class 9	1.000	0.527	1.000	0.563	-1.000	-0.524	-0.842	-0.435	-0.995	-0.523	-0.552	-0.320
	Class 10	1.000	0.505	1.000	0.565	-0.995	-0.503	-0.907	-0.464	-1.000	-0.503	-0.589	-0.322
	Class 11	1.000	0.505	1.000	0.565	-0.975	-0.501	-0.862	-0.416	-0.970	-0.501	-0.579	-0.313
	Class 12	1.000	0.560	1.000	0.561	-0.980	-0.545	-0.812	-0.417	-0.975	-0.544	-0.496	-0.310
	Class 13	1.000	0.587	1.000	0.559	-0.990	-0.584	-0.722	-0.406	-0.985	-0.584	-0.588	-0.337