SMU
SINGAPORE MANAGEMENT UNIVERSITY

School of
Computing and Information Systems

Master of Science in Computing

Capstone Report

Title of Project:

Harm Detection via Sparse Latent Representations in Large Language Models

Locating latent representations to detect 'harmful' behaviour in Large Language Models (LLMs) with sparse autoencoders and concept probes

1 December 2024
Lin Xin Rose
Supervised by: Professor Sun Jun

Table of Contents

Introduction

LLM Safety via harmful/adversarial prompt detection

With the rise of large language models (LLMs) and applications in society, the question of its safety and security becomes increasingly pertinent. A key area of concern for LLM (Large language model) safety is its generation of harmful text in response to harmful prompts. This concern is accentuated by the presence of adversarial prompts techniques to jailbreak existing strategies such as safety fine-tuning to safeguard these models [2, 3]. One of the approaches to mitigate this is via effective detection of these harmful / adversarial prompts when it is received by the model and hence preventing the LLM from generating harmful responses as early as possible. However, **how should we find an effective source of information for detecting these prompts?**

A direct source of information would be the prompts themselves. Competitions have supported the development of simple filtering or semantic detection techniques to filter harmful prompts [1]. Nevertheless, these methods remain limited given the boundless search space of possibilities to prompt a model. Additionally, these methods are further challenged by increasingly deceptive jailbreak attack prompts developed by red teaming efforts.

Harmful model behaviour detection via causal analysis of model internals

An alternative perspective to this problem could be to observe the impact of a prompt on the model's hidden representations. For instance, by detecting how prompt tokens affects model's logits or internal activations / representations when intervened upon (i.e. replaced with neutral token) [24]. This may be especially effective in cases where adversarial prompt tokens are generated based on optimisation techniques with respect to the model (eg. GCG algorithm) since these adversarial prompt tokens would inherently carry greater effect on model activations than normal prompts. This is also conceptualised as extracting the "causal distribution" of the prompt on model internals. Additionally, aside from analysing the prompts, [24] also used the difference in causal distributions across model layers (causal contribution of model layers on model logits) as a way to detect misbehaviour (i.e. jailbreak attempt detection, toxicity detection, bias detection) in language models.

Research Objective/Hypothesis: This project adds on to the overall goal to locate relevant latent representation of a model to detect harmfulness. Specifically this project aims to investigate the use of latent representations within a single model layer output.

(I) Can sparse representations detect harmful behaviour?

- First, it hypothesizes the effectiveness of sparsely disentangled representation of a single model layer to elicit meaningful fine grained latent representation space for effective and robust detection of harmful model behaviour. Sparse representation is gathered using pre-trained Sparse Auto-Encoders .

(II) Can localised sparse representations detect harmful behaviour?

- Second, it hypothesizes the effectiveness of localising concept relevant latents as a subset of the full sparse representation space to achieve and strengthen effective and robust detection of harmful model behaviour in a more efficient manner. Relevance of sparse latents is measured with respect to a concept probe in two ways, by correlation and causal scores.

Effectiveness is evaluated via classification metrics (Accuracy, Recall). Robustness is evaluated via recall on perturbed test data. Further details are described under the 'Experiments' section.

Related Works / Concepts

This section delineates the motivation and inspiration for the methods used in this project. Two emerging and increasing complementary subfields seeking to serve the goal to improve understanding and transparency of the internal workings of neural networks are mechanistic interpretability and representation engineering. A summary of their comparison is found in Table 1 below.

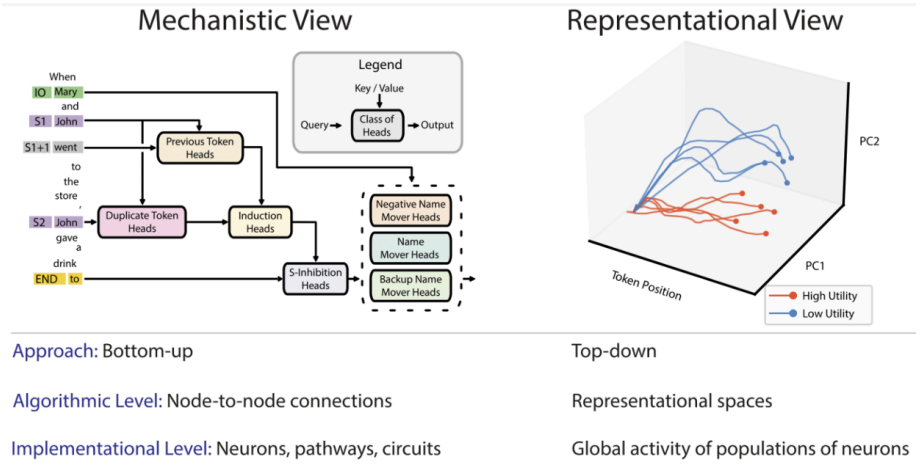


Diagram 1: Comparing the differences between Mechanistic Interpretability and Representation Engineering [6]

	Mechanistic Interpretability	**Representation Engineering / Model editing**
Existing Works and Insights	Bottom up perspective: unsupervised learning of features (eg. via sparse autoencoders) Shows discovery of feature circuits that identifies model components for performing linguistic tasks (eg. subject verb agreement) Related works: [15, 7, 4]	Top down perspective: supervised learning of features (eg. via 'vector' representation of a target concept) Shows promising results of model steering (i.e. changing model behaviour) for factual knowledge and concepts Related works: [6, 10]
Gaps	**Lesser concept-level feature analysis**: Lesser results have been shown for analysing conceptual level features (eg. harmfulness, truthfulness) **Need for quantitative metrics and use-case evaluations:** Most 'interpretability' insights remain qualitative, with a call for more quantitative measures and evaluation of results for down-stream applications [15]	**Black-box:** Less precise understanding of how the model represents and processes the concept vector

Table 1: Summary of comparison between Mechanistic Interpretability and Representation Engineering

Mechanistic Interpretability: Sparsity

Interpretability of Model mechanisms via Features and Circuits

Mechanistic interpretability hypothesizes that internal workings of deep models are mechanisms (i.e. computational processes) composed from fundamental units (i.e. features) and their relation/pathways with one another (i.e. circuits). Hence, techniques are developed with the goal to discover features, and potential circuits that explain model behaviour. For instance, sparse auto-enoders and causal tracing / attribution techniques are two approaches to disentangle the representation space and approximate feature circuits respectively [4].

Sparse Auto-Encoders as approximation of Dictionary Learning (Compressed Sensing)

The field started with neurons as initial fundamental units of analysis. However, it was observed that neurons are often polysemantic. They activates for multiple semantically different inputs suggesting that they may capture multiple semantic features at the same time. This was hypothesized as a property of "superposition" where neural networks represent more feature dimensions than neuron dimensions and that they do so by using a linear combination of neurons [22]. This insight is similar to approaches from Representation Engineering.

Hence, taking inspiration from Dictionary Learning from compressed sensing, learning a sparse representation via Sparse AutoEncoders (SAE) is seen as an approximation to finding the dictionary bases via Dictionary Learning [7]. In Dictionary Learning, dictionary bases are the key for effectively transforming sparse signal representations back into a full image. Hence, the goal is that SAE latents may be trained to approximate dictionary bases of the input feature space and reveal latent representations of interpretable and monosemantic features of the input space. Alternatively, this is viewed as effectively decomposing and disentangling superposition representation into its individual feature dimensions. Intuitively, this can be understood in Diagram 2. However, optimal training and evaluation for these SAEs can be a challenge given the primarily qualitative nature of interpretability as its goal. It is currently an active area of research [7, 8, 9].

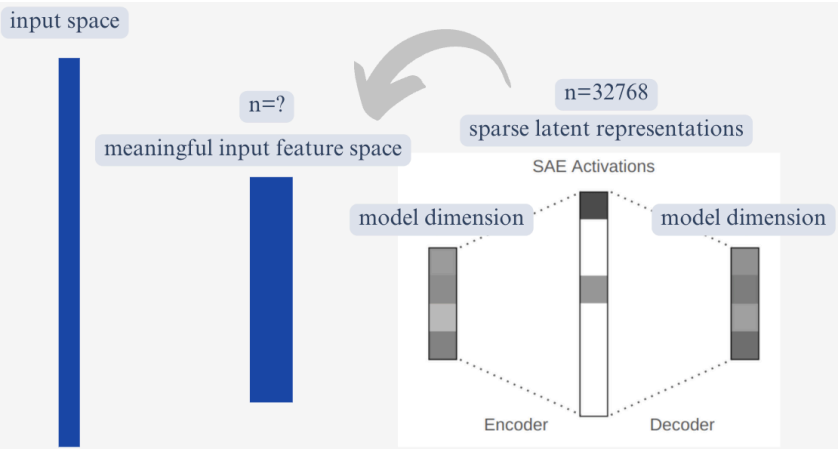


Diagram 2: Illustrating sparse latent representation as an approximation of the unknown input feature space.

Additionally, in compressed sensing, sparsity has been related to robust feature extraction for downstream classification. Hence, this project similarly hypothesizes the robustness of using sparse representation for harm detection.

Mostly evaluated on NLP based task behaviour
However, evaluation of the techniques developed in these fields have been primarily on traditional NLP based tasks (e.g. subject verb agreement, indirect object identification) or arithmetic reasoning tasks [7]. These tasks generally rely on lower-level and syntactic language features. Hence, may be seen as mainly evaluating for sequential patterns or structures. On the other hand, harmful text generation goes beyond syntactical / task based language features but may be seen as a higher level conceptual feature. In other words, the problem of harmful text generation would more likely concern model internal representations of high level features of concepts (eg. "harm") rather than low task features. Hence, this project aims to evaluate its application for high level feature representation.

Representation Engineering/NLP Analysis: Concept probing via linear classifier

Meanwhile, representation engineering studied neural networks with the hypothesis of representation spaces as units of analysis. Techniques are developed and evaluated for high level concept control, considering concepts as an unsupervised vector representation direction [6] or as embedding within a classifier [10]. However, these techniques generally do not consider interpretability of representation attributed back to the input space. Nevertheless, probing, a traditional NLP interpretability analysis technique, takes on similar assumptions of using linear representation spaces, using linear classifiers for characterising latent representations of a language model [12].

Similarly, this project takes inspiration from these works to embed the concept of “harm” within a linear representation space of a linear classifier. Although, whilst [6] located concepts via a crafted design template, in this project, the linear concept probe is simply trained over a set of harmful and harmful datasets.

Causality Analysis as a Framework: Neural network as a causal graph

Causality Analysis for Model Security Evaluation
Beyond acting as a 'feature' for detecting harmful model behaviour mentioned earlier, causality techniques can be more generally seen as a framework for studying and analysing neural networks by viewing them as a causal graph. By doing so, [13, 14] measured the causal contribution of different model components (neurons, layers) to the generation of safe / unsafe model responses and observed greater contributions of early model layers in enabling safety detection whilst greater contribution of later layers for carrying toxic feature representations.

Causality for language model interpretability
Similarly, efforts towards language model interpretability and analysis also adopted the causal framework for studying and explaining language model behaviours. For instance, explaining model prediction mechanisms for various NLP tasks through identifying fine-grained causal sub-graphs (circuits) of the model based on "causal contributions" of respective model components [4] or creating high level causal graph abstractions to improve symbolic representation and control over these models' predictions [16, 17]. Overall, causality has been observed as a promising measure for studying model behaviours with respect to model internal as well as an effective source of information for detecting harmful model behaviours. In this project, causal analysis is adopted as a measure to locate relevant latent representations with respect to a concept probe. It is compared with correlation as a way to attribute relations between these representation spaces.

Methods

To evaluate the effectiveness of (I) sparse representation and (II) localised sparse representation for effective and robust harm detection, the following methods were implemented to answer the respective research objectives.

(I) Does sparse representation detect harm? To answer this we compare the performance of a classifier trained over sparse representation with a classifier simply trained over original model dimensions as a baseline
(II) Does localised sparse representation detect harm? To answer this we compare the performance of a classifier trained over subsets of sparse representation located by correlation or causal score with respect to sparse classifier of method 2.

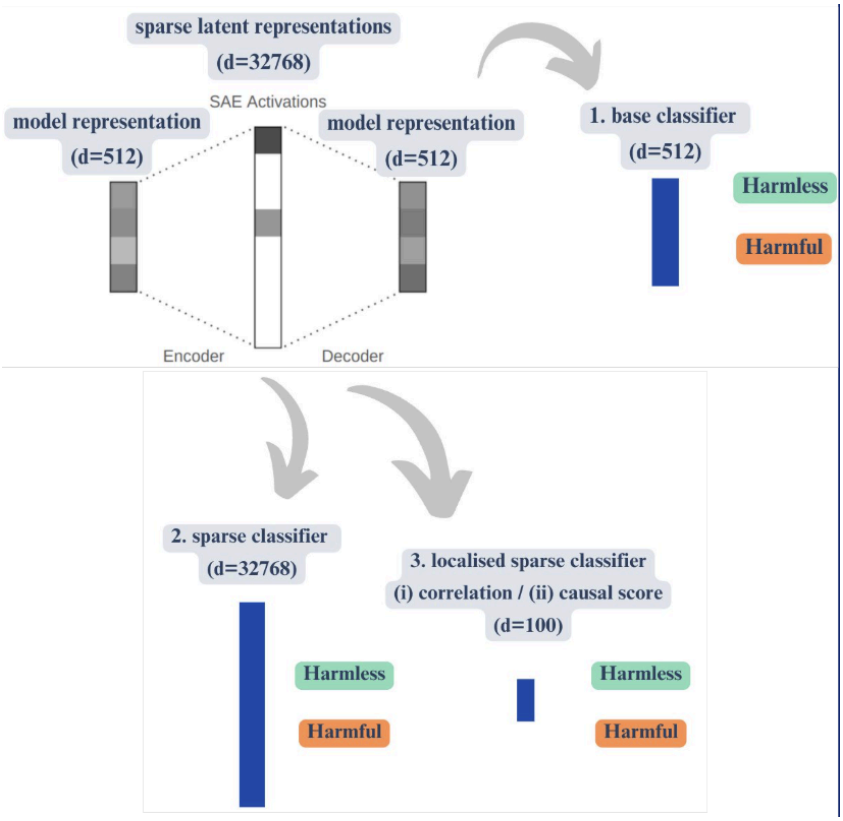


Diagram 3: Illustration of the methods investigated as described Table 3
Dimensions are based on experiment settings done in this project, as detailed under Experiment Setup

	Method	**Description | Motivation | Complexity**
1	Base Classifier (Baseline)	This method classifies over activations of model hidden representations directly. **Probing** This method is a NLP analysis technique used to characterise representations of a model layer based on its performance ability on targeted NLP tasks [11]. Hence, considered as a baseline measure. **Inference / Storage Complexity:** Dim = model dimensions

2	Sparse Classifier	This method classifies over sparse representation of model hidden representations. Sparse representations are gathered from activations of a pre-trained Sparse Auto-Encoder of the respective model layer. **Sparse latent representation as robust representations** Sparse Autoencoders are generally pretrained over model activations based on a large general set of data such as 'the Pile', as an approximate reconstruction of the sparse input feature space learnt by the model. For Pythia 70M this set of data corresponds to data used for its model training as well. Hence the sparse latent representations learnt by the sparse autoencoder is hypothesized as the disentangled feature space of the input data, 'the Pile'. With inspiration from compressed sensing, such a sparse representation may improve robustness of feature representations. **Inference / Storage Complexity: Higher than baseline** Dim = Sparse Autoencoder dimensions (e.g. 64 times of model dimensions for Pythia SAE. Despite its sparse activation, this can result in a much larger space and time complexity)
3	Localised Sparse Classifier (Correlation / Causal Score)	This method classifies over a subset of the sparse latent representation. **Localising most relevant sparse latents for parsimonious classification** Given that sparse latents are seen as a disentangled representation of the input feature space, localising seeks to find most relevant features for the classification (harm detection) task. Thereby improving efficiency by pruning less relevant features.
		Two methods for finding most relevant sparse latents are explored: **3.1 Localising based on most correlated latents:** Correlation values are measured by weights of the linear classifier which are the co-efficients of the corresponding sparse latents. A subset of k latent index corresponding to highest weight values are selected. See Figure 1. **3.1 Localising based on most causally contributing latents:** Causal values are measured by ablation effects (also understood as causal indirect effect) of the latent with respect to the sparse linear classifier. A subset of k latent index corresponding to highest ablation values are selected. See Figure 2. **Inference Complexity: Lower than baseline depending on size of k** Dim = Number of selected sparse latents (Top k)

Table 3: Implementation description of respective methods.
Complexity refers to Classifier Dimension/Num of Parameters.

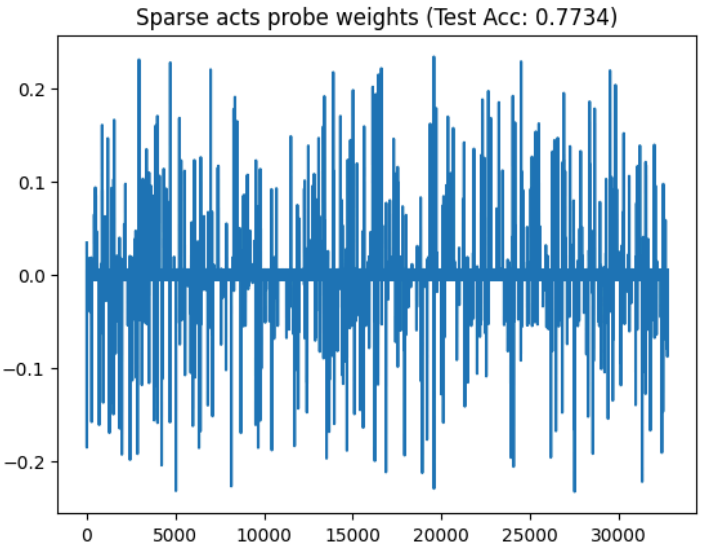


Figure 1: Learned weights for sparse classifier
Method to localise: Top k latents are select based on highest absolute weight values as a measure of correlation with concept probe trained to classify harmful and non-harmful prompts

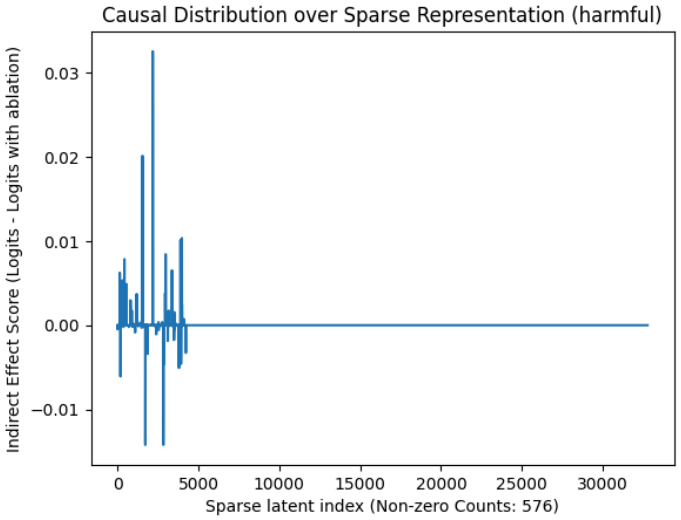


Figure 2: Ablation effects for each latent with respect to sparse classifier (Causal Indirect Effects based on logits differences in prediction harmful prompts)
Method to localise: Top k latents are select based on highest absolute ablation effects as a measure of causal contribution of the latent feature to concept probe for predicting harmful prompts

Experiment

Set up:

To investigate the effectiveness and robustness of the respective methods for detecting harmful model behaviour, the following experiment set up shown in Table 4 is used.

Model	Pythia 70M

Pretrained SAE layer	SAE Trained over Layer 4 Residual Stream Output [4] Accessed via python library: https://jbloomaus.github.io/SAELens/
Datasets: **Train, Test**	Set of balanced dataset taken from [6]. Training set (n=128): Harmful (n=64), Harmless (n=64) Test set (n=384): Harmful (n=192), Harmless (n=192) **Metrics computed:** Accuracy and Recall as a measure of Effective detection
Dataset: **Test Perturbed**	Harmful jailbreak prompts generated using autoDAN algorithm from [19] based on harmful prompts from test set (n=192) **Metrics computed:** Recall over perturbed data as a measure of Robustness

Table 4: Experiment Setup Details

Model Choice: With increasing release of pre-trained SAEs, future works may extend the experiment to bigger model sizes to evaluate the method on large scale large language models [21]. At the time of this project, Pythia 70M was chosen as a start given the pretrained weights released by [4]. Future works may also further extend these experiments to observe results on different sparse representations learnt by improved techniques for SAE training. Figure 3 observes the average sparse activation counts over the given dataset using Pythia.

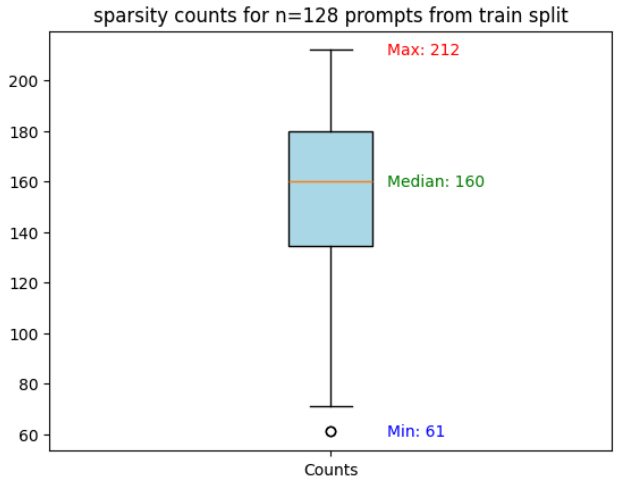


Figure 3: Average Sparse activation counts using SAE trained over Pythia
For understanding dimensions of the sparse representation (projection using SAE) of the original hidden dimensions of the model. Average range of activation counts is observed to be less than the original dimension size of 512, suggesting successful dimension reduction via sparsity

Layer choice: Layer 4 is chosen out of 6 layers in Pythia with reference to [4]. Additionally, middle layers of transformers are observed to carry greater semantic value for downstream tasks [20]. Deeper layers of BERT are also where features of greater complexity and high level features are learnt [11]. Additionally, [13, 14, 24] demonstrated the causal role of layers in detection of harmful behaviour. Future works may adopt layer wise scanning for toxicity/harm to better locate the target layer to train probe over.

Metrics for "Effectiveness" and "Robustness": Recall over test and jailbreak prompts

To measure “effectiveness” of detection: Accuracy and Recall over test set distribution, which is 3 times larger than the training set in this dataset as observed in Table 4.

To measure “robustness” of detection: Recall over perturbed harmful prompts from the test set is used. Perturbation is done by simply applying a jailbreak algorithm (eg. autoDAN) over the prompts using Pythia as the model for jailbreaking. Future works may further investigate robustness of the method to other jailbreak techniques to better understand its properties.

With autoDAN as perturbation, beyond difference in token length, some significant differences in sparse activation distribution is observed between original test distribution and the perturbed test distribution (Figure 5). Whereas differences in original activation distribution does not reveal significant differences in distribution. Full comparison of distribution between train, test and test perturbed data is shown in the Appendix.

Dataset EDA:

	Train	Test	Test Perturbed
Average Prompt Tokens length	12-13 tokens	12-13 tokens	61- 62 tokens

Table 5: Dataset EDA to understand differences in data split distribution

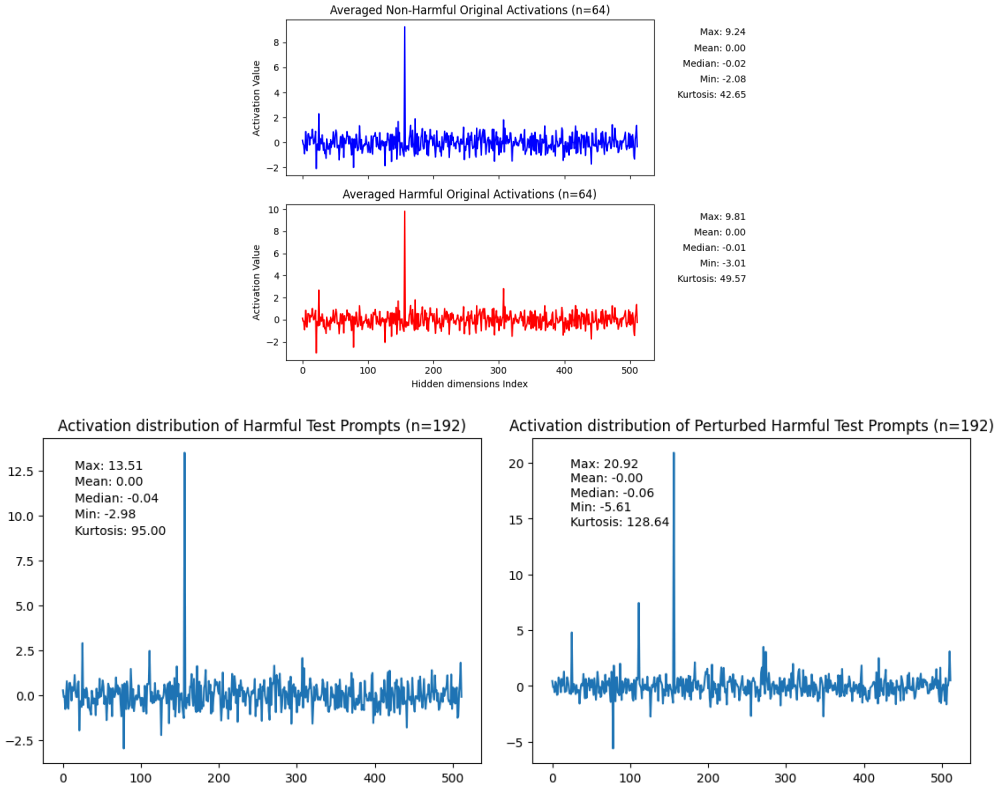


Figure 4: Comparing train (top), test (left) and perturbed (right) test distribution of model layer activations (d=512) *Distributions seem to be relatively similar.*

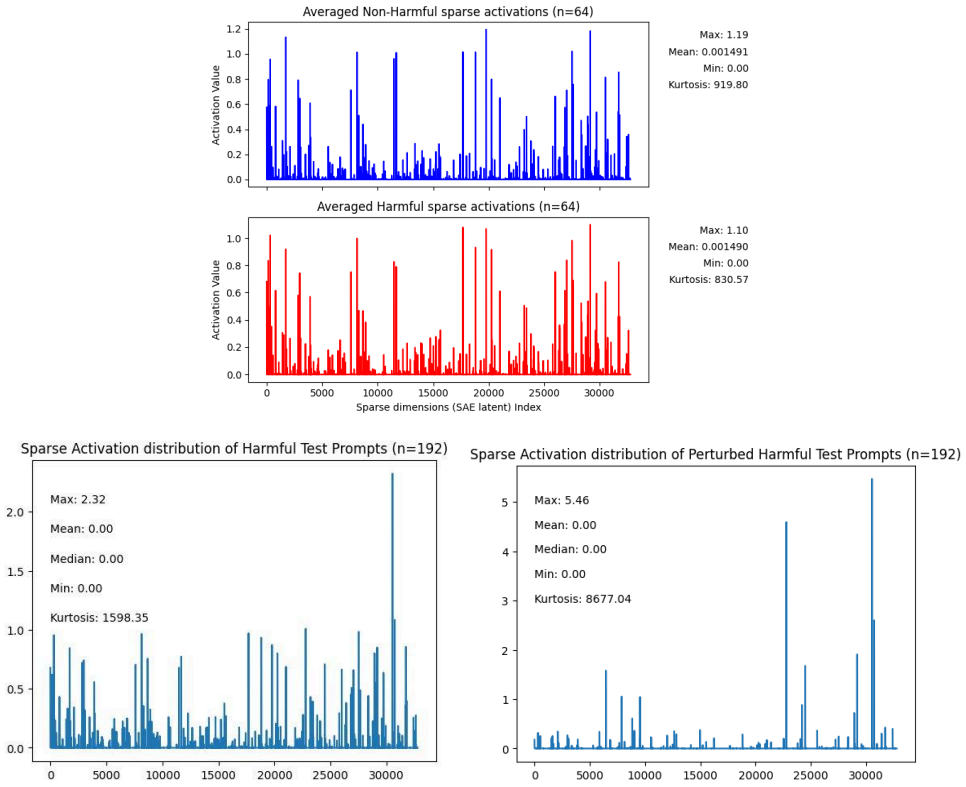


Figure 5: Comparing train (top), test (left) and perturbed (right) test distribution of model layer sparse activations (d=32768) *Sparse activation of perturbed data is significantly different. It seem to suggest distinct changes in activations elicited by the sparse representation of activations which are not elicited by original activation dimensions (Figure 4)*

(I) Can sparse representations detect harmful behaviour?

This section reports results with respect to the first research objective.

Qualitative: Training Data Distribution Plots

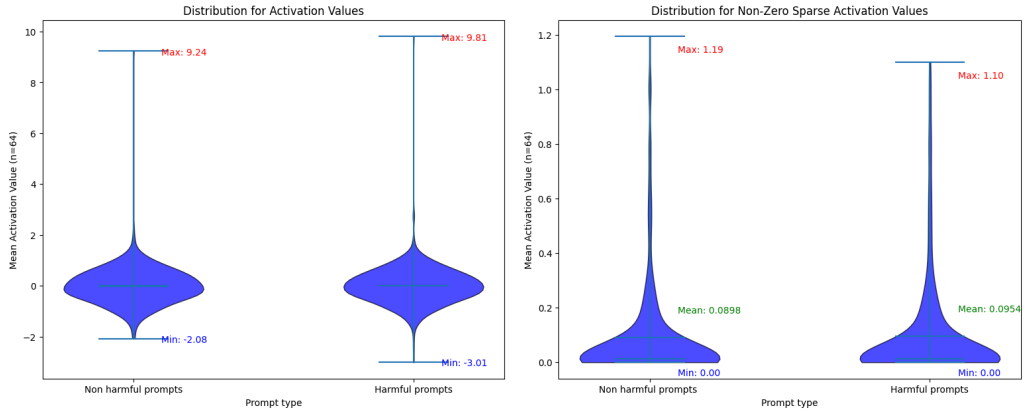


Figure 6: Violin plot of harmful vs non-harmful training data activation distributions *(Left) Violin plot of harmful vs non-harmful activation values (Right) Violin plot of harmful vs non-harmful sparse activation values (Plot is created based on non-zero activation values only) Differences in statistical distribution observed, however, there are no significant differences in shape*

Quantitative Metrics: (1) Base Classifier vs (2) Sparse Classifier

Epochs = 65, Train N=128	(1) Base	(2) Sparse	(2) Sparse (epochs =3)
Accuracy (N=384)	89.8%	73.2%	77.3%
Recall on Harmful prompts (N=192)	82.2%	69.3%	91.1%
Recall on Perturbed Harmful prompts (N=192)	0%	57.3%	79.7%

Table 6: Comparison of classification results for Base vs Sparse classifier

Classification over sparse representation performs better on Recall and Robustness, but poorer on Accuracy. References: probe_results.ipynb; Training plot under Appendix C

Quantitative: Sparse classifier perform better in Recall and Robustness. (Table 6)
Quantitative results show that base classifier performs better on accuracy, while sparse classifier performs better on recall for both test and test perturbed dataset. Given the use case of harm detection, recall performance can be weighted more important than accuracy. Additionally, better recall performance on the perturbed distribution suggests better robustness of the sparse classifier.
Qualitative: Sparse representation may elicit more differences in perturbed data (Figure 5). Although training distributions do not show significant differences in shape between harmful and non-harmful prompts. EDA analysis from Figure 5 observed that the sparse representation of perturbed distribution elicits more significant differences in activation patterns than original base representations. Hence, supporting the quantitative results of stronger robustness.

(II) Can localised sparse representations detect harmful behaviour?

Based on previous results, we have observed the potential value of using sparse representations. However, despite sparsity of values, computationally its large dimension requires greater space complexity to load and run, such complexity can be increased significantly with the increased scale of the language model as well. Hence, can we identify a subset of relevant sparse latents to detect harmful behaviour more efficiently? Relevance of sparse latents is measured with respect to the sparse acts concept probe in two ways, by correlation and causal scores

Experiment setup: The results were run over a set of top 100 latents located as a start. Future works can look to increase this number for performance improvements of the classifier. In this set up, results are compared to better understand the behaviour of different localising metrics.

Qualitative: Distribution Plots for localised sparse activations

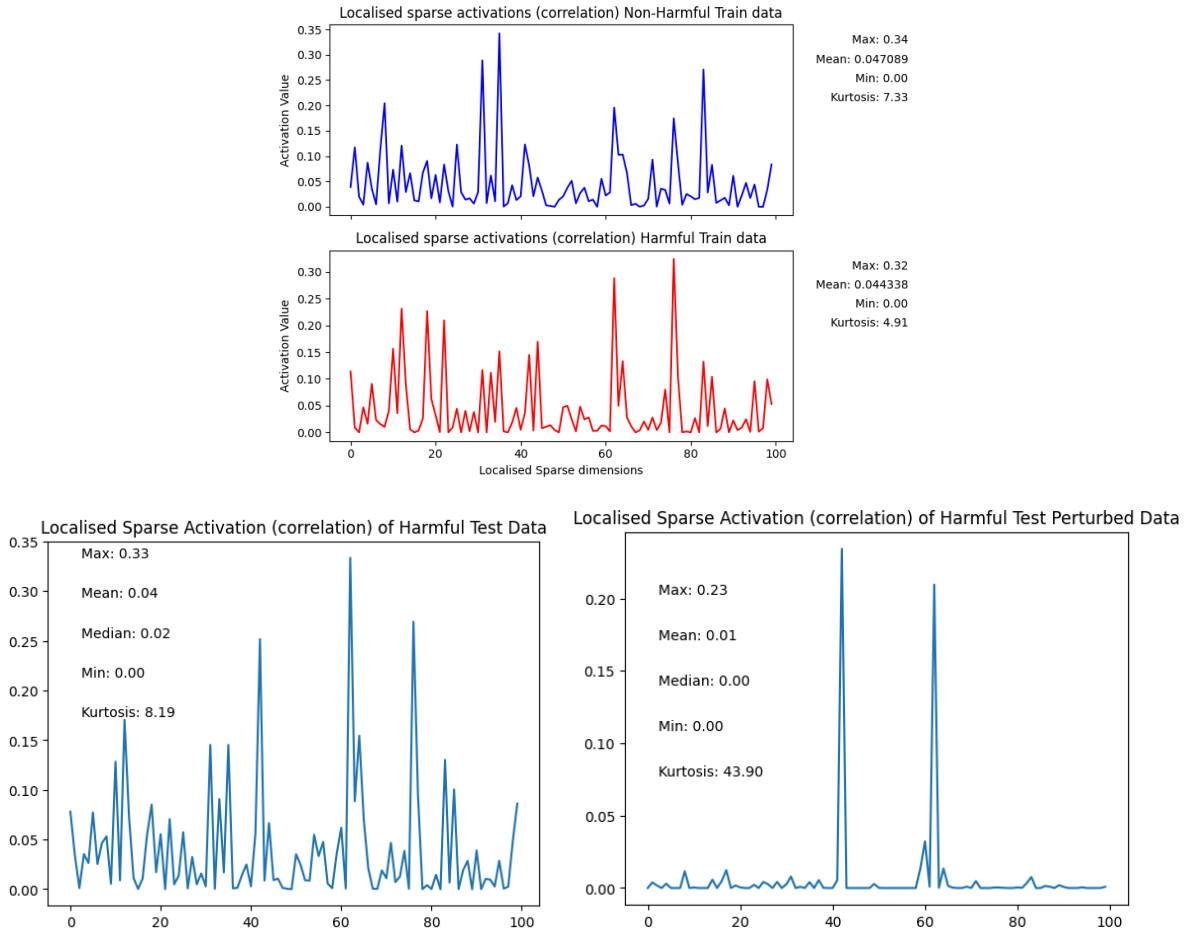
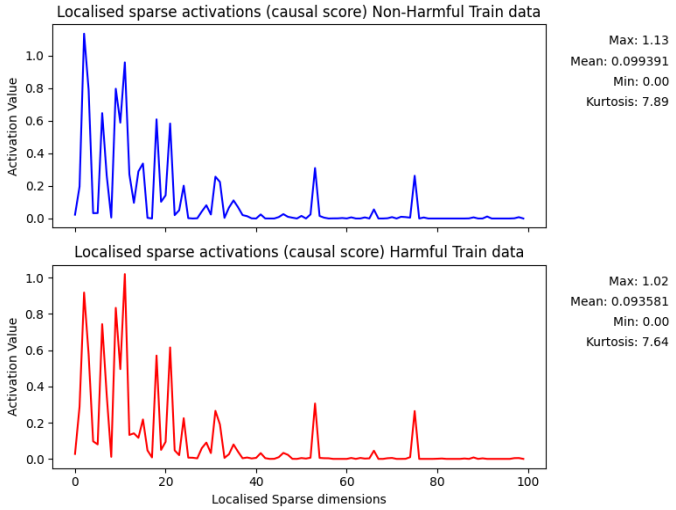


Figure 6: Sparse activation distribution of train (top), test (left), test perturbed (right) over top 100 latents by weight values (correlation)*, As expected perturbed data distribution is significantly different from original train and test distribution.*



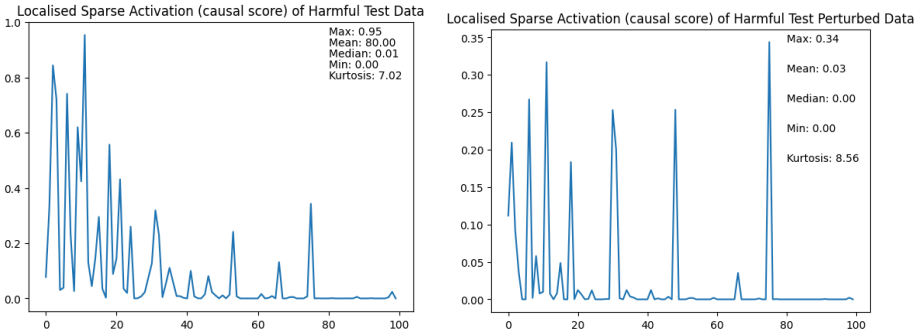


Figure 6: Sparse activation distribution of train (top), test (left), test perturbed (right) over top 100 latents by ablation effects (causal score)*. Similarly perturbed data distribution (right) is significantly different from original train (top) and test (left) distribution.*

Quantitative Metrics: (3) Localised Sparse Classifier by weights and ablation effect

Epochs = 65, train N=128 Topk / dim = 100	**Max Weights (correlation)**	**Max Ablation effect (causal score)**
Accuracy (N=384)	76.3%	58.9%
Recall (N=192) (Harmful Prompts only)	59.3%	55.2%
Recall on Perturbed Harmful (N=192) **(**Harmful prompts only)	0%	34.9%

Table 6: Comparison of classification results for over top 100 latents localised by weights and ablation effect with respect to sparse classifier (of 77% accuracy)

Qualitative: Distribution plots over localised sparse activations are as expected where representations over train and test distributions are similar while test perturbed (jailbreak testset prompts) show a rather different distribution. With this EDA, we move to the quantitative results.
Quantitative: For "Accuracy": Latents located by correlation are observed to perform better than latents located by causal score. For "Robustness": Latents located by causal score are observed to perform better than latents located by correlation. Nevertheless, classification performance over the lower dimension of 100 latents is poor, a higher number of subset latents can be explored to improve performance.

Qualitative: Interpretability of latents located by weights and ablation effect

Given that SAEs have been developed for improving monosemanticity hence interpretability of language models. This analysis seeks to ask: **Does most relevant latents correspond to harm related features from the input space as well?** Interestingly, this is not completely so, especially not for causality located latents. This might suggest correlation as a better measure for locating concept relevant latents.

Top 5 latents	By Weight values	By Ablation effect
Top activating input prompts	**Related:** UV rays, LGBT discrimination, vulgar words **Not related:** Button Toggling Code, "Potential" token	**Related:** NIL **Not related:** 4 non-interpretable latents Japan related

Table 7: Interpretability of top 5 latents located by weight (correlation) and ablation effect (causal score) with respect to sparse classifier (of 77% accuracy). Screenshots of maximally activating tokens are found in Appendix

Conclusion / Discussion

Overall, this project gained following conclusions to the respective research objectives:
(I) Does sparse latent representation improve harm detection? Although the classifier over sparse representation performed poorer than baseline on test set accuracy, it performed better on perturbed distribution suggesting better robustness than a simple classification over hidden activations. Distribution plots over sparse representation also reveal more distinct differences between original test and perturbed test distribution than original hidden representations. Therefore, yes, sparse latent representation demonstrate potential to improve robust harm detection.

(II) How well does a localised sparse classifier perform? Similarly, results from both localisation methods do not perform as well as baseline on the test distribution. However, causally localised sparse classifier seem to show greater robustness than baseline. Nevertheless, given its low test accuracy performance and the poor training plot of a causal sparse probe (see Appendix C). Further experiments on using causality to localise more robust feature latents can be done for better evaluation of its effects.

Nevertheless, these initial results reveal the possibility of harm detection via representations of a single model layer. Notably, it suggests a potential for improving robust harm detection via sparse representation and causally located latent representations of layer activations. Future works could further investigate the potential of these properties for developing robust methods via further experimentations on different jailbreak techniques and model sizes. This method may also adopt methods from [14] to select targeted model layers with high toxicity measures to improve the extraction of effective, robust and relevant latent representations for harm detection.

References

[1] “Large Language Models Capture-the-Flag (LLM CTF) Competition,” *Spylab.ai*, 2024. https://ctf.spylab.ai/

[2] A. Zou, Z. Wang, K. J. Zico, and M. Fredrikson, “Universal and Transferable Adversarial Attacks on Aligned Language Models,” *arXiv.org*, 2023. https://arxiv.org/abs/2307.15043

[3] L. Weng, “Adversarial Attacks on LLMs,” *lilianweng.github.io*, Oct. 25, 2023. https://lilianweng.github.io/posts/2023-10-25-adv-attack-llm/

[4] S. Marks, C. Rager, E. J. Michaud, Y. Belinkov, D. Bau, and A. Mueller, “Sparse Feature Circuits: Discovering and Editing Interpretable Causal Graphs in Language Models,” *arXiv.org*, 2024. https://arxiv.org/abs/2403.19647

[5] K. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt, “Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 small,” *arXiv.org*, Nov. 01, 2022. https://arxiv.org/abs/2211.00593

[6] A. Zou *et al.*, “Representation Engineering: A Top-Down Approach to AI Transparency,” *arXiv.org*, Oct. 10, 2023. https://arxiv.org/abs/2310.01405

[7] T. Bricken, “Towards Monosemanticity: Decomposing Language Models With Dictionary Learning,” *transformer-circuits.pub*, Nov. 04, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html

[8] L. Gao *et al.*, “Scaling and evaluating sparse autoencoders,” *arXiv.org*, Jun. 06, 2024. https://arxiv.org/abs/2406.04093

[9] S. Rajamanoharan *et al.*, “Improving Dictionary Learning with Gated Sparse Autoencoders,” *arXiv.org*, 2024. https://arxiv.org/abs/2404.16014

[10] Y. Zhang, Z. Wei, J. Sun, and M. Sun, “Adversarial Representation Engineering: A General Model Editing Framework for Large Language Models,” *arXiv.org*, 2024. https://arxiv.org/abs/2404.13752 (accessed Dec. 01, 2024).

[11] I. Tenney, D. Das, and E. Pavlick, “BERT Rediscovers the Classical NLP Pipeline,” 2019. Accessed: Sep. 07, 2024. [Online]. Available: https://fq.pkwyx.com/default/https/aclanthology.org/P19-1452.pdf

[12] C. Potts, “Analysis methods in NLP: Probing CS224u: Natural language understanding.” Accessed: Dec. 01, 2024. [Online]. Available: https://web.stanford.edu/class/cs224u/2021/slides/cs224u-2021-analysis-part4-handout.pdf

[13] “Causality Analysis for Evaluating the Security of Large Language Models,” *Arxiv.org*, 2023.

https://arxiv.org/html/2312.07876v1 (accessed Dec. 01, 2024).

[14] W. Zhao, Z. Li, Y. Li, Y. Zhang, and J. Sun, “Defending Large Language Models Against Jailbreak Attacks via Layer-specific Editing,” pp. 5094–5109, Jan. 2024, doi: https://doi.org/10.18653/v1/2024.findings-emnlp.293.

[15] T. Räuker, A. Ho, S. Casper, and D. Hadfield-Menell, “Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks,” *arXiv.org*, Aug. 18, 2023. https://arxiv.org/abs/2207.13243

[16] A. Geiger *et al.*, “Causal Abstraction: A Theoretical Foundation for Mechanistic Interpretability,” *arXiv.org*, 2023. https://arxiv.org/abs/2301.04709 (accessed Dec. 01, 2024).

[17] A. Geiger, Z. Wu, C. Potts, T. Icard, and N. D. Goodman, “Finding Alignments Between Interpretable Causal Variables and Distributed Neural Representations,” *arXiv.org*, 2023. https://arxiv.org/abs/2303.02536 (accessed Dec. 01, 2024).

[18] “Sparse Autoencoders for Pythia-70M-Deduped,” *Neuronpedia.org*, 2024. https://www.neuronpedia.org/p70d-sm (accessed Dec. 01, 2024).

[19] X. Liu, N. Xu, M. Chen, and C. Xiao, “AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models,” *arXiv.org*, Oct. 03, 2023. https://arxiv.org/abs/2310.04451

[20] L. Valeriani, D. Doimo, F. Cuturello, A. Laio, A. Ansuini, and A. Cazzaniga, “The geometry of hidden representations of large transformer models,” *arXiv.org*, Oct. 30, 2023. https://arxiv.org/abs/2302.00294

[21] “Neuronpedia,” *Neuronpedia*, 2024. https://www.neuronpedia.org/ (accessed Dec. 01, 2024).

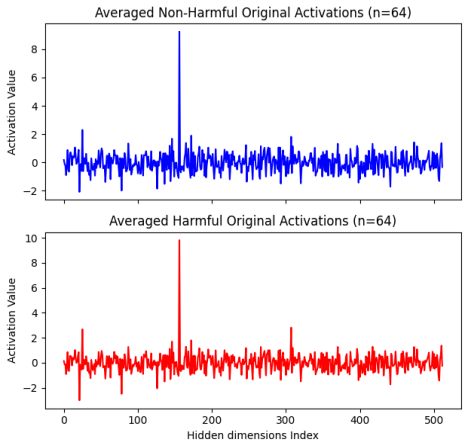
[22] “Toy Models of Superposition,” *transformer-circuits.pub*. https://transformer-circuits.pub/2022/toy_model/index.html

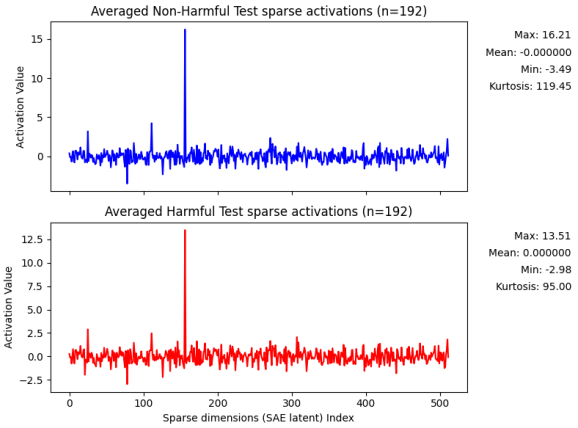
[23] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, “Locating and Editing Factual Associations in GPT,” *arXiv:2202.05262 [cs]*, Jan. 2023, Available: https://arxiv.org/abs/2202.05262

[24] M. Zhang, K. K. Goh, P. Zhang, and J. Sun, “LLMScan: Causal Scan for LLM Misbehavior Detection,” *arXiv.org*, 2024. https://arxiv.org/abs/2410.16638 (accessed Dec. 01, 2024).

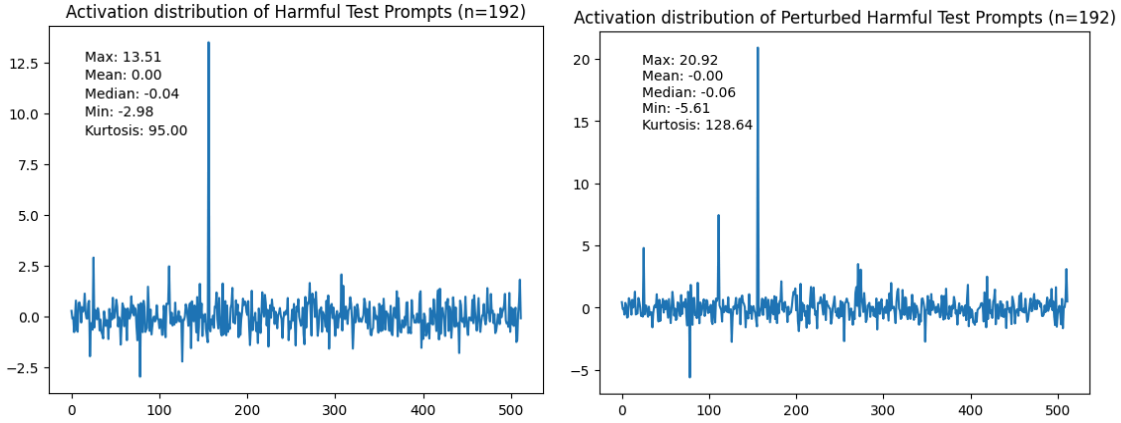
Appendix

A. EDA over original activations distribution (train, test, test perturbed)





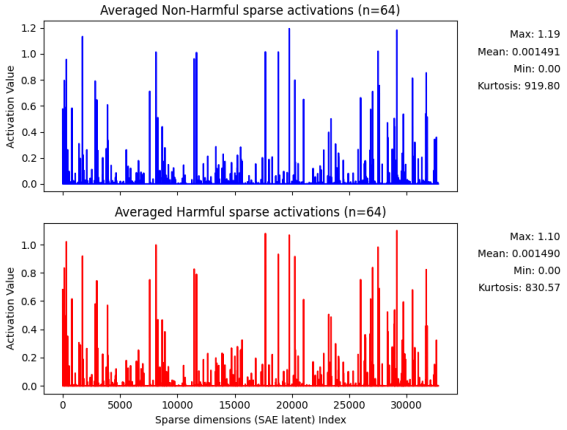
Train vs Test Distributions

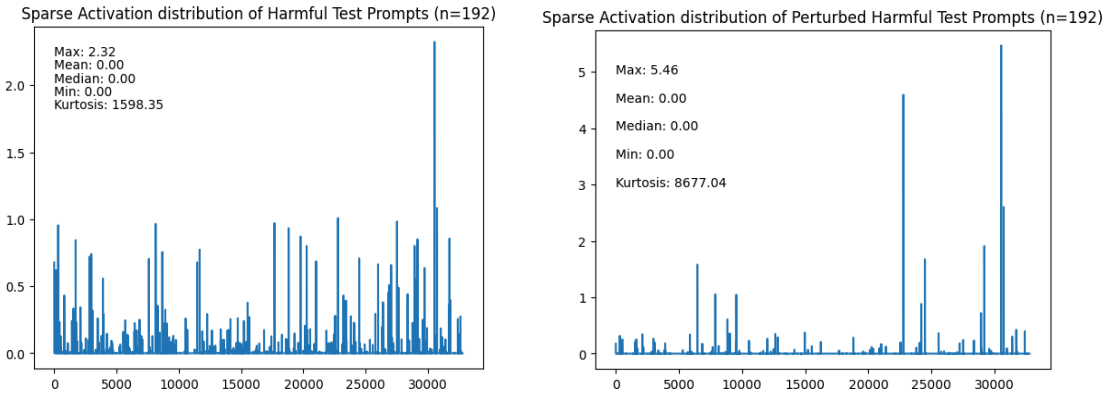


Test vs Test Perturbed

Distributions remain relatively similar.

B. Dataset: EDA over train, test and test perturbed sparse activations

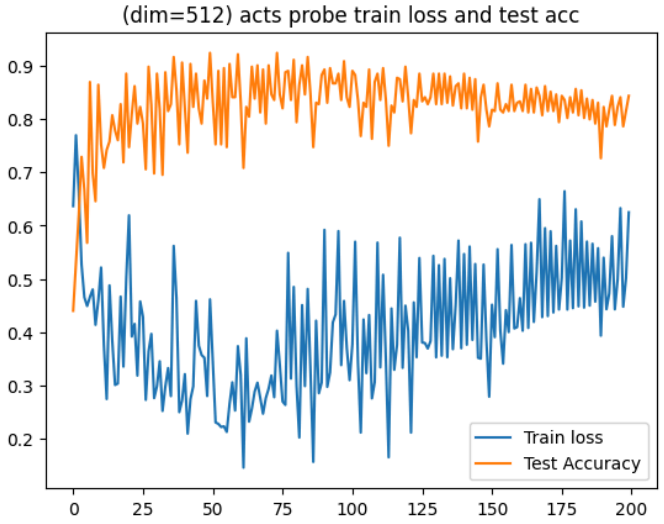




Comparing sparse activation distributions of train, test and test perturbed dataset.

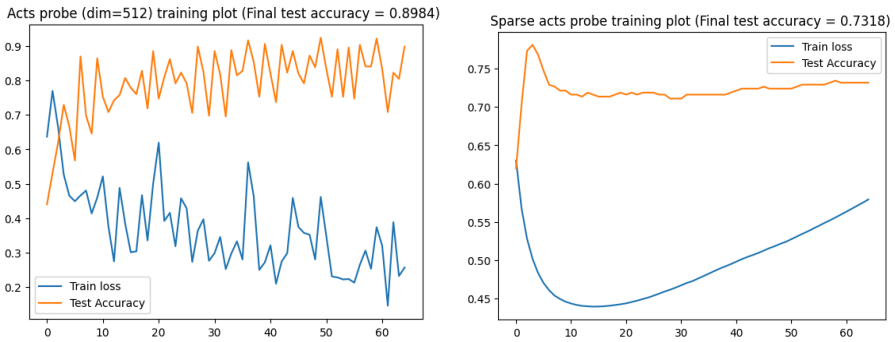
Sparse activation distribution of test set remains relatively similar to training set compared to the Sparse activation of perturbed test distribution is significantly different as observed by the kurtosis score (a measure of distribution dispersion), with a larger range of activation values as well.

C. Training plots



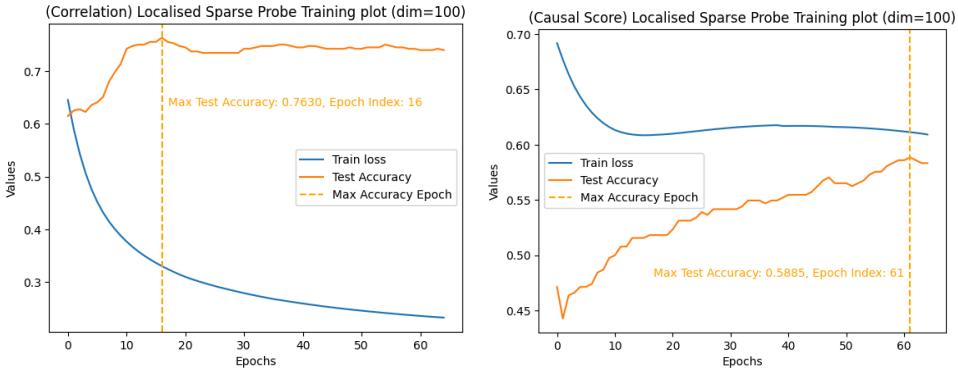
Linear classifier (baseline) training over 200 epochs to decide optimal training epoch

Based on the results, epochs = 65 was chosen as default training setting, given highest test accuracy.



Training plot of Original probe vs Sparse probe

Epcohes = 65 was chosen as it gave highest test accuracy for baseline linear classifier (Appendix B)



Training plot of Sparse classifier over top 100 latents by correlation and causal score

Training process for causally localised sparse probe can be improved

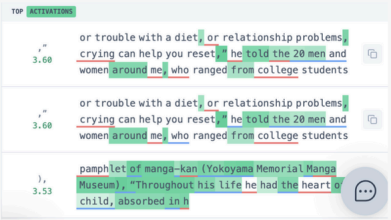
Future works may consider more than 100 latents for better training results.

D. Results for Objective 2: Interpretability of located latents

II Results: Locate by concept probe - interpretability

552 (not meaningful)

2977 (not meaningful)



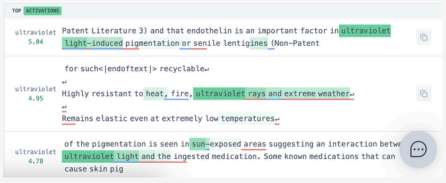
319 (japan related)

3898 (not meaningful)

2180 (not meaningful..)

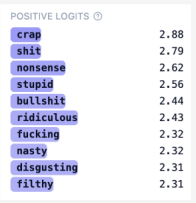
causal	**552, 2977, 319*(japan related), 3898, 2180**

II Results: Locate by concept probe - interpretability



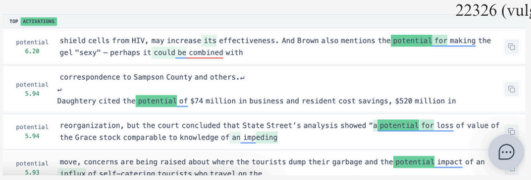
6332 (UV rays: bio-harm)

14630 (LGBT discrimmination)



22326 (vulgar words)

29218 (enable/disable toggling)



13347 ("potential")

correlational	6332, 29218, 14630, 13347, 22326 (based on absolute values)