Semantic Entropy Neurons: Encoding Semantic Uncertainty in the Latent Space of LLMs

Jiatong Han[1†*] **Jannik Kossen**[1] **Muhammed Razzak**[1] **Yarin Gal**[1]

Abstract

Uncertainty estimation in Large Language Models (LLMs) is challenging because token-level uncertainty includes uncertainty over lexical and syntactical variations, and thus fails to accurately capture uncertainty over the semantic meaning of the generation. To address this, Farquhar et al. [10] have recently introduced semantic uncertainty (SE), which quantifies uncertainty in the semantic meaning by aggregating token-level probabilities of generations if they are semantically equivalent. Kossen et al. [18] further demonstrated that SE can be cheaply and reliably captured using linear probes on the model hidden states. In this work, we build on these results and show that semantic uncertainty in LLMs can be predicted from only a very small set of neurons. We find these neurons by training linear probes with L_1 regularization. Our approach matches the performance of full-neuron probes in predicting SE. An intervention study further shows these neurons causally affect the semantic uncertainty of model generations. Our findings reveal how hidden-state neurons encode semantic uncertainty, present a method to manipulate this uncertainty, and contribute insights for the field of interpretability research.

1 Introduction

Estimating uncertainty in texts generated by Large Language Models (LLMs) is challenging because token-level uncertainty measures can be biased by variations in syntactic or lexical forms, even when the underlying meaning remains the same. Semantic uncertainty addresses this by focusing on uncertainty within the space of semantic meaning [10, 19]. Concretely, Farquhar et al. [10] sample multiple model generations, cluster them by their semantic meanings, and compute the entropy over the aggregated probabilities of the semantic clusters. Due to the high computational costs of this sampling-based approach, recent work has focused on linearly probing semantic entropy from the hidden states, a technique that has proven effective in both in-distribution and out-of-distribution scenarios [18].

In this work, we investigate the hypothesis that semantic uncertainty is a feature linearly represented by only a small subset of neurons in the hidden states [9]. This aligns with studies identifying neurons responsible for concepts like toxicity [21] and unsafe features rooted in the hidden states of LLMs [39]. Following Kossen et al. [18], we train linear probes to predict SE from the latent space using a training set consisting of activation values of model generations and the corresponding SE scores. We use the probe weights to assess the importance of individual neurons in contributing to semantic uncertainty during model generation. By applying strong L_1 regularization, we induce sparsity in the probes, allowing us to identify key neurons—referred to as **semantic entropy neurons**—that are most predictive of semantic uncertainty in the hidden layers of LLMs. To confirm semantic entropy neurons causally impact the SE of model generations, we study the *counterfactual* effects of clamping the activation of these neurons during inference, inspired by Templeton et al. [39]. Our findings reveal that clamping causally impacts semantic entropy, i.e. we can control the level of SE by clamping the neurons, although it does degrade model calibration. We further demonstrate that on a

[*] Correspondence to jiatong.han@u.nus.edu, {jannik.kossen,muhammed.razzak}@cs.ox.ac.uk. [1] OATML, Department of Computer Science, University of Oxford. [†] Work done while at OATML.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

text completion task, Indirect Object Identification [44], clamping significantly affects the likelihoods of semantically relevant tokens, unlike tempering which affects all tokens indiscriminantly.

In summary, our key contributions are:

- We demonstrate that a small set of SE neurons is sufficient to match the full SE probing performance that uses the entire set (2,048 times more) of hidden space neurons (Section 4).
- We show that clamping individual neurons to activation values corresponding to high- or low-SE model generations effectively manipulates semantic entropy in the intended directions. Additionally, we show that the effects of clamping these neurons are distinct from generating with increased model temperatures (Section 5).

2 Background

Semantic entropy (SE) is an effective measure of semantic uncertainty that can be used to detect hallucinations in LLMs [10], where models generate plausible-sounding but factually incorrect outputs [27, 11, 16]. SE is measured over clusters of semantically equivalent outputs, where two generations are semantically equivalent if they entail each other bi-directionally. Probes—lightweight classifiers trained on hidden states—have proven effective in predicting various linguistic properties or future tokens [2, 35]. Kossen et al. [18] have shown that SE is a robust supervisory signal for training semantic entropy probes (SEPs) on the model hidden states, thus providing a cost-efficient method for quantifying semantic uncertainty and detecting hallucinations in LLM generations.

3 Semantic Entropy Neurons

Sparsity-Induced Neurons. We hypothesize that the hidden space neurons, $h_l^p(x)$, for token position p at model layer l given input x, provide an over-complete basis for representing semantic entropy, with only a few neurons being sufficient to distinguish between high and low semantic entropy generations. We refer to these neurons as semantic entropy neurons, h_{SE}^p. We train linear probes with L_1 regularization (i.e., Lasso regression) to encourage sparsity. With a sufficiently high L_1 penalty, only a small number of neurons retain non-zero coefficients. If our hypothesis is true, the L1 penalty should not negatively impact probing performances. To ensure consistent neuron selection across model layers, we pool individual neurons with the highest absolute linear weights from the layer-wise probes to form the set of SE neurons. We consider p to be the second-last generated token given input x, as shown to be effective for probing in [18].

Activation Clamping. If linearly probing with a limited number of SE neurons yields strong performance in predicting SE, then the activations of SE neurons should display linearly separable distribution gaps across high and low semantic entropy values. To determine whether these differences are causally affecting the SE in model generations (rather than being spurious correlations), we use a counterfactual approach. Following Kossen et al. [18], we first divide the model generations into high and low SE clusters by dividing along the best-split SE threshold. We then compute the maximum (or minimum) neuron activations for the high (or low) SE sample clusters (see Eq. (1)). We manually clamp the neuron activations for inputs in each cluster to the activation values of the opposite cluster, i.e., clamping neurons of high SE inputs to the low SE activation values and vice versa. The clamped activation for $\hat{h}_{\text{SE}}^p(x)$ in layer l given input x and the set of all model generations X' can be formulated as

$$\hat{h}_{\text{SE=A}}^p(x) = \text{sign}\left(h_{\text{SE=A}}^p(x)\right) \cdot \begin{cases} \max_{x' \in X'}\left(|h_{\text{SE=B}}^p(x')|\right), & \text{if } A = \text{Low}, B = \text{High} \\ \min_{x' \in X'}\left(|h_{\text{SE=B}}^p(x')|\right), & \text{if } A = \text{High}, B = \text{Low}, \end{cases} \tag{1}$$

where $A, B \in \{\text{High, Low}\}$ and $B \neq A$. In preliminary experiments, we observe that the activation gap $|h_{\text{SE=high}}^p(x) - h_{\text{SE=low}}^p(x)|$ for input x often peaks at the layer l where the probe's performance is optimal. This is expected since this should be where the linear classification boundary lies. Therefore, we clamp at the layer where the probes perform best: $l = \text{argmax}_{l'}(T_{l'})$, where $T_{l'}$ is the test AUROC for probes trained on the activation of layer l'. Following Nanda [30], we apply clamping at a single layer in the residual stream to enable fine-grained causal intervention.

We evaluate the differences in likelihoods of generating the first answer token when the model is prompted with a question x, with or without low→high activation clamping in the short-form generation settings [18]. Let $P_t(y, x)$ denote the probability of generating token y at temperature t when answering x, and $P'_t(y, x)$ the probability under activation clamping. We wish to show:

$$\frac{1}{|Y_1|}\sum_{y\in Y_1}\frac{P'_t(y,x)-P_{t'}(y,x)}{P_t(y,x)} > \max\left\{\frac{1}{|Y_2|}\sum_{y\in Y_2}\frac{P'_t(y,x)-P_{t'}(y,x)}{P_t(y,x)},\quad 0\right\},\quad \forall t' > t. \tag{2}$$

Intuitively, higher model temperatures (t') flatten the probability distribution, increasing the probabilities of most tokens, regardless of their *semantic relevance*. In contrast, activation clamping should selectively boost the probabilities of semantically relevant tokens, rather than uniformly affecting all token probabilities like tempering. Hence we check, in Equation (2), if the semantically relevant tokens (Y_1) are seeing proportionally greater probability boosts than semantically irrelevant tokens (Y_2), as well as than those under high tempering (i.e., l.h.s. of Eq. (2) > 0). Empirically, we find that Y_1 are usually the set of tokens that already has substantial probability mass before our intervention.

4 Single Neurons Capture Semantic Entropy

In this section, we compare the performance of predicting SE in-distribution using probes trained on full activations (SEPs) versus those trained only on SE neurons (Sparse SEPs). Refer to Appendix B for details on experiment setup. We pool SE neurons across datasets to ensure consistent inputs for Sparse SEPs (cf. Section 3) and find that only 2 or 3 neurons out of 4,096 or 8,192 are selected (Tab. 8). See Figure 1 for layer-wise performance comparisons. We find that sparse SEPs either outperform or match SEPs across all datasets and models, particularly in mid-to-late layers. These results confirm our hypothesis that hidden space neurons provide an over-complete basis for representing semantic uncertainty (Section 3).

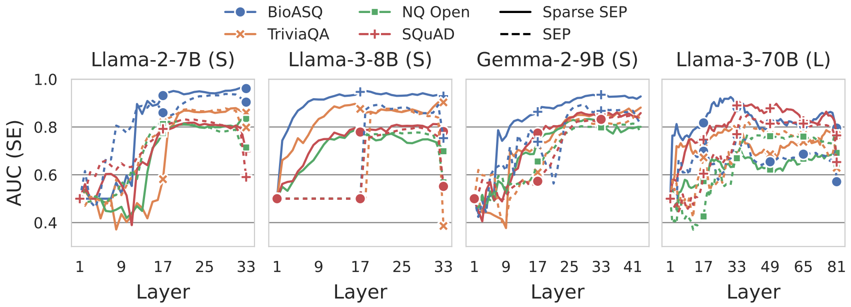


Figure 1: Sparse SEPs using only a few neurons (≤ 3) match or outperforms SEPs using thousands of neurons (≥ 4096). Layer-wise probing performance comparison between full activation (SEPs) and SE neurons activation (Sparse SEPs) across models. Tested for in-distribution SE prediction.

We additionally show probes trained on SE neurons perform well in out-of-distribution (OOD) tests. As shown in Table 7 (right), Sparse SEPs match the performances of SEPs. This suggests that SE neurons exhibit activation patterns generalizable across datasets, similar to those captured by SEPs trained on full activations. See Appendix B for all evaluation details.

We investigate if semantic entropy neurons are polysemantic [5], or account for more fine-grained characteristics of semantic uncertainty so as to be domain-generalizable. Our preliminary findings (see Appendix D.5) show that these neurons strongly activate in response to tokens on topics where models are more likely to make factual errors, such as numerical data, citations, or security codes (cf. Tab. 9).

5 Activation Clamping Manipulates SE

Clamping SE Neurons Controls SE. We perform activation clamping on the Llama-3-8B and Gemma-2-9B models. We find that activation clamping has a significant impact on the semantic entropy of model generations in the intended directions (see Tab. 3 and Fig. C.2). For example, under high→low SE clamping, SE reduced by 35.6% on average across datasets for Llama-3-8B, and it was 27.1% for Gemma-2-9B. This implies semantic entropy neurons are causally responsible for the changes in SE. We further study if clamping causally affects model accuracy since SE can be used to detect hallucinations. In Table C.2, we show that the overall model accuracy after clamping remains largely unaffected, which droped at most by 1.17%. However, we find clamping could let model behave in less calibrated ways, e.g. being more confidently wrong or ambiguously correct, which we elaborate in Appendix C.

Model	$\Delta_{\text{SE/Acc., Clp. Dir.}}$	**Sample**$_{\textbf{Incrt.}}$	**Sample**$_{\textbf{Crt.}}$
Llama-3-8B (S)	$\Delta_{\text{SE, high}\rightarrow\text{low}}$	$-0.286_{\pm 0.011}$	$-0.350_{\pm 0.023}$
	$\Delta_{\text{Acc., high}\rightarrow\text{low}}$	$+0.050_{\pm 0.006}$	$-0.200_{\pm 0.021}$
Gemma-2-9B (S)	$\Delta_{\text{SE, high}\rightarrow\text{low}}$	$-0.286_{\pm 0.011}$	$-0.409_{\pm 0.023}$
	$\Delta_{\text{Acc., high}\rightarrow\text{low}}$	$+0.041_{\pm 0.005}$	$-0.146_{\pm 0.018}$

Table 1: Mean and standard error of absolute SE changes and Accuracy (Acc.) changes over datasets for *previously* correct (Sample$_{\text{Crt.}}$) and incorrect (Sample$_{\text{Incrt.}}$) samples under high→low activation clamping. SE ranges from 1.055 to 2.303 before clamping. See low→high results in Table 2.

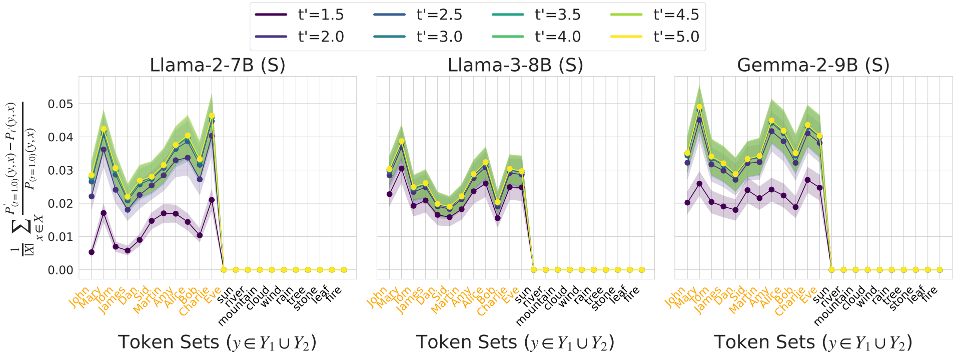


Figure 2: Activation clamping encourages more semantically sensible outputs than model tempering. Semantically relevant tokens (in orange) in response to the set of synthesized IOI prompts X experience a greater proportional increase in probabilities (cf. Eq. (2)) under clamping compared to high tempering and that of irrelevant tokens (in black) on average. Tested in (S)hort-form generation settings.

Logit Difference. We explore if activation clamping behaves differently from a simple tempering of the next token distribution. Increasing the temperature of the next token probability distribution increases the probability of tokens indiscriminately. In this section, we investigate if clamping behaves in a more semantically meaningful way. We evaluate this on a text completion task of Indirect Object Identification (IOI) [44]. We synthesize 500 IOI prompts (X) based on the IOI templates from Wang et al. [44]. We create a pool of 12 human names, 10 locations, and 30 relevant items to fill up the template, and ensure they fit appropriately for each context. An example template is "After the lunch, `[A]` and `[B]` went to the `[LOCATION]`. John gave a `[ITEM]` to", and a filled prompt can be "After the lunch, John and Martin went to the shops. John gave a bag to", where " Martin" is more expected to be the model completion. We categorize the 12 human names as *semantically relevant* (Y_1) and 10 irrelevant items separately generated (such as "sun", "river", "mountain") as *semantically irrelevant* (Y_2). We confirm that these categorizations correspond to higher or lower logits, with significant differences between the categories, from the generations of unclamped models (see Tab. 5). We share all details in Appendix D.2.

In Figure 2, we present the average proportional changes in token likelihoods (as computed in Eq. (2)) when asking models to complete the IOI prompts. The shady area is the standard error among the 500 sample completions. We observe that semantically relevant tokens are more frequently returned with low → high activation clamping ($t = 1$) than with high tempering ($t' > 1$), as indicated by positive y-axis values. Also, the semantically relevant tokens gain relatively more likelihood than the semantically irrelevant ones, conforming to the inequality in Equation (2). These findings suggest that activation clamping effectively adjusts the logits in a semantically reasonable manner, distinguishing it from model tempering, which uniformly affects all token probabilities regardless of their relevance as outputs.

6 Conclusion

In this work, we show that semantic entropy, as a measure of semantic uncertainty, can be reliably captured from linearly probing on just 2 or 3 neurons from the model hidden states (cf. Tab. 8). We further show that clamping the activation of these neurons can effectively manipulate the SE of model generations in the intended directions, while possibly making the model less calibrated. One future direction is to understand SE neurons by investigating their polysemanticity and finding interpretations through their activating patterns with input tokens [5].

References

[1] Azaria, A. and Mitchell, T. The internal state of an llm knows when it's lying. In *EMNLP*, 2023.

[2] Belinkov, Y. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 2021.

[3] Belrose, N., Furman, Z., Smith, L., Halawi, D., Ostrovsky, I., McKinney, L., Biderman, S., and Steinhardt, J. Eliciting latent predictions from transformers with the tuned lens, 2023. URL `https://arxiv.org/abs/2303.08112`.

[4] Bills, S., Cammarata, N., Mossing, D., Tillman, H., Gao, L., Goh, G., Sutskever, I., Leike, J., Wu, J., and Saunders, W. Language models can explain neurons in language models. `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`, 2023.

[5] Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL `https://transformer-circuits.pub/2023/monosemantic-features`. Transformer Circuits Thread.

[6] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *NeurIPS*, 2020.

[7] Cunningham, H. and Conerly, T. Comparing topk and gated saes to standard saes, Jun 2024. URL `https://transformer-circuits.pub/2024/june-update/index.html#topk-gated-comparison`.

[8] Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[9] Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. Toy models of superposition, 2022. URL `https://arxiv.org/abs/2209.10652`.

[10] Farquhar, S., Kossen, J., Kuhn, L., and Gal, Y. Detecting Hallucinations in Large Language Models Using Semantic Entropy. *Nature*, 2024.

[11] Filippova, K. Controlled hallucinations: Learning to generate faithfully from noisy data. In *EMNLP*, 2020.

[12] Gemma, T. Gemma. *Kaggle*, 2024. doi: 10.34740/KAGGLE/M/3301. URL `https://www.kaggle.com/m/3301`.

[13] Goh, G. Decoding the thought vector, 2016. URL `https://gabgoh.github.io/ThoughtVectors/`.

[14] Halawi, D., Denain, J.-S., and Steinhardt, J. Overthinking the truth: Understanding how language models process false demonstrations, 2024. URL `https://arxiv.org/abs/2307.09476`.

[15] He, P., Liu, X., Gao, J., and Chen, W. Deberta: Decoding-enhanced bert with disentangled attention. In *ICLR*, 2021.

[16] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

[17] Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. *ACL*, 2017.

[18] Kossen, J., Han, J., Razzak, M., Schut, L., Malik, S., and Gal, Y. Semantic entropy probes: Robust and cheap hallucination detection in llms, 2024. URL `https://arxiv.org/abs/2406.15927`.

[19] Kuhn, L., Gal, Y., and Farquhar, S. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *ICLR*, 2023.

[20] Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Kelcey, M., Devlin, J., Lee, K., Toutanova, K. N., Jones, L., Chang, M.-W., Dai, A., Uszkoreit, J., Le, Q., and Petrov, S. Natural questions: a benchmark for question answering research. *TACL*, 2019.

[21] Lee, A., Bai, X., Pres, I., Wattenberg, M., Kummerfeld, J. K., and Mihalcea, R. A mechanistic understanding of alignment algorithms: A case study on dpo and toxicity, 2024. URL https://arxiv.org/abs/2401.01967.

[22] Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. *NeurIPS*, 36, 2024.

[23] Lieberum, T., Rajamanoharan, S., Conmy, A., Smith, L., Sonnerat, N., Varma, V., Kramár, J., Dragan, A., Shah, R., and Nanda, N. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2, 2024. URL https://arxiv.org/abs/2408.05147.

[24] Lin, J. and Bloom, J. Announcing neuronpedia: Platform for accelerating research into sparse autoencoders, Mar 2024. URL https://www.lesswrong.com/posts/BaEQoxHhWPrkinmxd/announcing-neuronpedia-as-a-platform-to-accelerate-research.

[25] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

[26] Marks, S. and Tegmark, M. The Geometry of Truth: Emergent Linear Structure in Large Language Model Representations of True/False Datasets. *arXiv 2310.06824*, 2023.

[27] Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. On faithfulness and factuality in abstractive summarization. In *ACL*, 2020.

[28] Meta. Introducing meta llama 3: The most capable openly available llm to date, 2024. URL https://ai.meta.com/blog/meta-llama-3/. [Online; accessed June 16 2024].

[29] Nanda, N. A comprehensive mechanistic interpretability explainer & glossary, Dec 2022. URL https://neelnanda.io/glossary.

[30] Nanda, N., 2023. URL https://www.neelnanda.io/mechanistic-interpretability/attribution-patching.

[31] Nanda, N., Rajamanoharan, S., Kramar, J., and Shah, R. Fact finding: Attempting to reverse-engineer factual recall on the neuron level, Dec 2023. URL https://www.alignmentforum.org/posts/iGuwZTHWb6DFY3sKB/fact-finding-attempting-to-reverse-engineer-factual-recall.

[32] Nostalgebraist. Interpreting gpt: The logit lens, 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

[33] Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

[34] OpenAI. GPT-4 technical report, 2023.

[35] Pal, K., Sun, J., Yuan, A., Wallace, B., and Bau, D. Future lens: Anticipating subsequent tokens from a single hidden state. In *CoNLL*, 2023.

[36] Rajamanoharan, S., Conmy, A., Smith, L., Lieberum, T., Varma, V., Kramár, J., Shah, R., and Nanda, N. Improving dictionary learning with gated sparse autoencoders, 2024. URL https://arxiv.org/abs/2404.16014.

[37] Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for squad. *ACL*, 2018.

[38] Subramani, N., Suresh, N., and Peters, M. E. Extracting latent steering vectors from pretrained language models. *ACL*, 2022.

[39] Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N. L., McDougall, C., MacDiarmid, M., Freeman, C. D., Sumers, T. R., Rees, E., Batson, J., Jermyn, A., Carter, S., Olah, C., and Henighan, T. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

[40] Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[41] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023.

[42] Tsatsaronis, G., Balikas, G., Malakasiotis, P., Partalas, I., Zschunke, M., Alvers, M. R., Weissenborn, D., Krithara, A., Petridis, S., Polychronopoulos, D., Almirantis, Y., Pavlopoulos, J., Baskiotis, N., Gallinari, P., Artiéres, T., Ngomo, A.-C. N., Heino, N., Gaussier, E., Barrio-Alvers, L., Schroeder, M., Androutsopoulos, I., and Paliouras, G. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 2015.

[43] Turner, A. M., Thiergart, L., Leech, G., Udell, D., Vazquez, J. J., Mini, U., and MacDiarmid, M. Activation addition: Steering language models without optimization, 2024. URL `https://arxiv.org/abs/2308.10248`.

[44] Wang, K., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small, 2022. URL `https://arxiv.org/abs/2211.00593`.

[45] Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., et al. Representation engineering: A top-down approach to ai transparency. *arXiv:2310.01405*, 2023.

A Related Work

Understanding Hidden States. Recent research highlights the significance of LLM hidden states in influencing model behavior. Operations on these states can alter model outputs, including factual accuracy [45, 38]. Probes—lightweight classifiers trained on hidden states—have proven effective for predicting various linguistic properties and even future tokens [2, 35]. Notably, recent studies suggest that certain directions in the latent space correspond to the "truthfulness" of model outputs [26, 1].

Model Steering. Prior works such as [13, 33, 8, 31] seek to find the internal workings of neural networks by interpreting individual neurons and their interactions in deciding model behavior. Olah et al. [33] suggests that neural networks develop legible internal representations of features, which can be connected to form interpretable circuits. These features are causally meaningful variables that can be leveraged to steer the model, much like steering vectors. Meanwhile, it has been demonstrated that model behaviors can be steered by adding a vector to the model hidden states [22, 43], derived by calculating the differences in activation averages between specific model behavioral classes, similar to the activation clamping approach we will employ.

Logit Lens. The logit lens is a tool for examining how predictions develop within neural networks, particularly transformers like GPT, by directly multiplying intermediate layer activations with the model's unembedding matrix, which maps the model hidden states to the vocabulary space, to generate token logits for predicting the next token [14, 3, 32]. Due to the residual structure of transformers, the network tends to maintain a consistent basis across layers. It hence often reveals that prediction distributions converge toward the final output well before the last layer [32]. While the logit lens generally provides valuable insights into the model's decision process, Belrose et al. [3] argue that it only offers a biased view on a fraction of information encoded in the network. We build on the logit lens to precisely identify how changes in specific neuron activations influence the logits of ground truth and other relevant tokens (Section 3), a technique referred to as the logit difference method [29].

B Experiment Setup

Probing Semantic Entropy & Evaluation. We evaluate SEPs across four models based on their ability to capture semantic entropy. For short-form generations, we use Llama-2-7B [41], Llama-3-8B [28], and Gemma-2-9B [12], with DeBERTa-Large [15] as the entailment model. For long-form generations, we use Llama-3-70B [28] with GPT-3.5 [6] predicting entailment. Sparse SEPs are trained using Lasso regression [40] with the inverse of L_1 regularization constraint being 0.01. SEPs are evaluated on four QA datasets: TriviaQA [17], SQuAD [37], BioASQ [42], and NQ Open [20]. We compute the area under the receiver operating characteristic curve (AUROC), with ground truth labels given by binarized SE [18]. We ensure consistency with Kossen et al. [18] in dataset splits, model instructions, and the calculation of SE.

In additional to layer-wise training, we train Sparse SEPs on concatenated layers, and test them in the In-Distribution (ID) and Out-of-Distribution (OOD) tests in predicting SE. We follow the same layer concatenation strategy as in the hallucination detection experiments in Kossen et al. [18, Appendix B.3]. The concatenation details are shared in Table 6. Sparse SEPs are trained with a regularization parameter of $1/\lambda_1 = 0.01$. SEPs are trained on the same set of concatenated layers, allowing for a meaningful comparison since SEPs utilize a superset of information compared to Sparse SEPs. In the OOD tests, particularly, we train SEPs on one dataset and test them on others, and on each dataset, we report the average test performances using probes trained on others.

Activation Clamping. We further our experiments to identify semantic entropy neurons and manipulate them within the hidden states of Llama-3-8B and Gemma-2-9B for short-form generations. Activation clamping is performed on 500 samples from high- or low-SE cluster across multiple datasets. Additionally, we explored the logit differences using 500 synthesized Indirect Object Identification (IOI) examples referencing Wang et al. [44, Appendix E.].

C Activation Clamping

Additional Results. We show the observable gaps between the mean activation values of SE categories in Figure C.1. We report full clamping outcomes for Llama-3-8B and Gemma-2-9B in the short-form generation settings at Table 3. We present the absolute changes in SE and accuracy for different correctness classes before and after clamping in Table 2. We show the kernel density plots for Llama-3-8B after clamping at Figure C.2. We provide some concrete examples on how clamping would change model behaviors at high temperatures ($t = 1.0$) in Table 4.

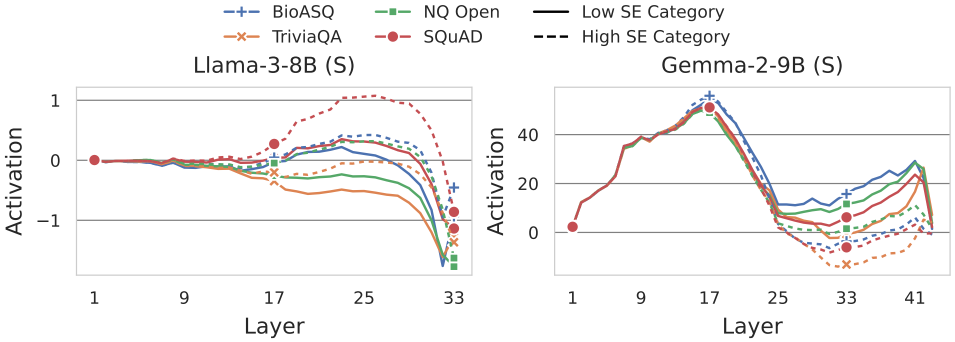
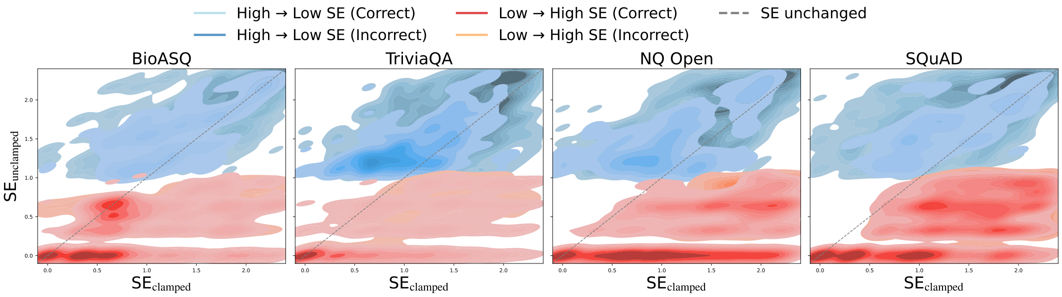


Figure C.1: Average SE neurons activation over samples for different SE categories across layers. There exist noticeable gaps between activation averages of the two categories. Llama-3-8B and Gemma-2-9B in (S)hort-form generation settings.

Model	$\Delta_{\text{SE/Acc., Clp. Dir.}}$	$\textbf{Sample}_{\textbf{Incrt.}}$	$\textbf{Sample}_{\textbf{Crt.}}$
Llama-3-8B (S)	$\Delta_{\text{SE, low}\rightarrow\text{high}}$	$+0.910_{\pm 0.022}$	$+2.392_{\pm 0.017}$
	$\Delta_{\text{Acc., low}\rightarrow\text{high}}$	$+0.046_{\pm 0.008}$	$-0.052_{\pm 0.006}$
Gemma-2-9B (S)	$\Delta_{\text{SE, low}\rightarrow\text{high}}$	$+0.457_{\pm 0.020}$	$+1.113_{\pm 0.011}$
	$\Delta_{\text{Acc., low}\rightarrow\text{high}}$	$+0.066_{\pm 0.010}$	$-0.025_{\pm 0.004}$

Table 2: Mean and standard error of absolute SE changes and Accuracy (Acc.) changes over datasets for *previously* correct ($\text{Sample}_{\text{Crt.}}$) and incorrect ($\text{Sample}_{\text{Incrt.}}$) samples under low→high activation clamping from Llama-3-8B and Gemma-2-9B models. SE ranges from 0.0 to 1.03 before clamping. The standard error indicates the variations among sample-wise changes in each measure and category of samples. See high→low results in Table 1.

Clamping Affects Model Calibration. We observe this in Table 1: under high→low SE clamping, the SE decreases by 0.35, but the accuracy drops by 0.20 on previously correct samples (both are absolute changes for Llama-3-8B). Under low→high SE clamping (see Tab. 2), the SE increases by 0.91, and yet the accuracy increases by 0.04 on previously incorrect samples. The results suggest that clamping could affect model faithfulness in counterfactual ways, where depending on the model correctness and SE *a priori*, model can be more confidently wrong, or more ambiguously correct.



	BioASQ	**TriviaQA**	**NQ Open**	**SQuAD**
High SE Samples				
Accuracy After Clamping	0.1800	0.2271	0.2060	0.1800
(Δ)	(↓0.0020)	(↓0.0160)	(↓0.0040)	(↑0.0220)
Mean SE After Clamping	1.5937	1.4537	1.3493	0.0000
(Δ)	(↓0.2277)	(↓0.1718)	(↓0.3287)	(↓1.8160)
Low SE Samples				
Accuracy After Clamping	0.7100	0.7260	0.5840	0.4900
(Δ)	(↓0.0100)	(↓0.0220)	(↓0.0120)	(↓0.0240)
Mean SE After Clamping	0.8869	0.7181	1.4111	1.4071
(Δ)	(↑0.5415)	(↑0.5164)	(↑1.0545)	(↑0.9508)

Figure C.2: (Top) Kernel density plots of SE changes after clamping in two directions: low → high (red) or high → low (blue) clamping. Correctness is determined by the answers of *unclamped* Llama-3-8B in the short-form generation setting. (Bottom) Table C.2: Changes in LLM correctness and SE for Llama-3-8B in (S)hort-form Generation Setting after activation clamping. Table entries are results after clamping.

Table 3: Changes in LLM accuracy and semantic entropy (SE) during model generations for Llama-3-8B and Gemma-2-9B in the (S)hort-form generation setting, following activation clamping on SE neurons for samples with low or high SE. The table entries show the values after clamping. For both models, activation clamping hardly affects model accuracy while significantly changing semantic entropy.

Llama-3-8B (S)

	BioASQ	**TriviaQA**	**NQ Open**	**SQuAD**
Low SE Samples				
Accuracy	0.72 → 0.71 ↓	0.75 → 0.73 ↓	0.60 → 0.58 ↓	0.50 → 0.50
Mean SE	0.35 → 0.89 ↑	0.20 → 0.72 ↑	0.36 → 1.41 ↑	0.46 → 1.41 ↑
High SE Samples				
Accuracy	0.22 → 0.18 ↓	0.28 → 0.27 ↓	0.20 → 0.21 ↑	0.14 → 0.16 ↑
Mean SE	1.75 → 1.51 ↓	1.66 → 1.28 ↓	1.67 → 1.36 ↓	1.72 → 1.40 ↓

Gemma-2-9B (S)

	BioASQ	**TriviaQA**	**NQ Open**	**SQuAD**
Low SE Samples				
Accuracy	0.74 → 0.74	0.80 → 0.80	0.60 → 0.60	0.50 → 0.50
Mean SE	0.29 → 0.48 ↑	0.19 → 0.43 ↑	0.33 → 0.68 ↑	0.39 → 0.75 ↑
High SE Samples				
Accuracy	0.22 → 0.21 ↓	0.28 → 0.27 ↓	0.20 → 0.21 ↑	0.14 → 0.16 ↑
Mean SE	1.75 → 1.51 ↓	1.66 → 1.28 ↓	1.67 → 1.36 ↓	1.72 → 1.40 ↓

Table 4: Selected generations ($t = 1.0$) using activation clamping from Llama-3-8B on TriviaQA and NQ Open in the short-form generation setting. The clamping does influence generation consistency and hence semantic entropy (SE).

Question	**Answer**	**Generations**	**Clamped Generations**
		High → Low Clamping on TriviaQA	
Who won the 2014 FIFA World Cup?	Germany	Argentina, Brazil, Netherlands	Germany, Germany, Germany
Which actor played Captain America?	Chris Evans	Robert Downey Jr., Chris Pratt, Chris Hemsworth	Chris Evans, Chris Evans, Chris Evans
What is the capital of France?	Paris	Berlin, Madrid, Rome	Paris, Paris, Paris
What year did World War II end?	1945	1941, 1942, 1944	1945, 1945, 1945
		Low → High Clamping on NQ Open	
Who created an engine using high pressure steam in 1801?	Oliver Evans	Richard Trevithick, Richard Trevithick, Richard Trevithick	French engineer Richard Trevithick, British engineer Richard Trevithick, Matthew Murray
French troops put down the Camisard uprisings between what years?	1702 and 1709	1702 and 1710, 1702-1710, 1702-1710	1722 and 1724, 1666 and 1699, 1701 and 1710
Who was a prominent Huguenot in Holland?	Pierre Bayle	William the Silent, William the Silent, William the Silent	Admiral Michiel de Ruyter, Willem Usselincx, William the Silent of Orange
What was the name of Watt's partner?	Boulton	Boulton, Boulton, Boulton	Boulton, and James B. & Wyl Wilson & John Muir, John Buss and Matthew Murray

D Experiment Details

Here we provide details to reproduce the experiments of the main body of this paper.

D.1 Prompt Templates

We adhere to the prompt templates outlined in Kossen et al. [18, Appendix B.1] for guiding both short- and long-form model generations.

D.2 Synthesis of IOI Samples

Name Lists. We generate 12 random people names, 10 random locations, 30 objects—each set of 3 objects associated with one location—and 10 additional words that would never appear in the samples. The people names are treated as semantically relevant tokens, while the extra words are considered semantically irrelevant tokens. All the generated names are presented below.

```
# Define lists of possible names, objects, and places
# To fill in [A], [B], [PLACE], and [OBJECT] in the IOI templates.
names = ["John", "Mary", "Tom", "James", "Dan", "Sid", "Martin", "Amy",
"Alice", "Bob", "Charlie", "Eve"]
places = ["shops", "park", "office", "restaurant", "cinema", "beach",
"library", "museum", "airport", "cafe"]
objects = {
    "shops": ["bag", "apple", "book"],
    "park": ["ball", "kite", "frisbee"],
    "office": ["document", "pen", "notebook"],
    "restaurant": ["drink", "menu", "receipt"],
    "cinema": ["ticket", "popcorn", "drink"],
    "beach": ["shell", "towel", "sunscreen"],
    "library": ["book", "magazine", "newspaper"],
    "museum": ["souvenir", "map", "postcard"],
    "airport": ["passport", "boarding pass", "luggage"],
    "cafe": ["coffee", "sandwich", "cake"]
}
extra_words = ["sun", "river", "mountain", "cloud", "wind", "rain",
"tree", "stone", "leaf", "fire"]
```

Logit Differences. We measure the logit differences between the semantically relevant and irrelevant token categories, as presented in Table 5. We find that the logits for the semantically relevant category are significantly higher than those for the irrelevant category, with differences exceeding two standard errors for each pair of tokens considered from the two categories.

Sample Generation from Templates. We use GPT-4o to generate the name lists and complete IOI samples with the below prompt:

```
[IOI Templates]

1. Generate Lists:
   - Generate 20 unique single-word names, evenly split by presumed gender.
   - Generate 10 unique single-word place names.
   - Generate 10 unique single-word object names.

2. Sample Generation:
   - Generate 500 samples using the IOI templates provided.
   - Randomly fill in the missing fields ([A], [B], [PLACE], [OBJECT])
   using the generated lists.
   - Ensure that [A] and [B] are always different.
   - Each template, name, place, and object should be selected with
   equal probability.
```

Semantically Relevant Tokens	Llama-2-7B	Llama-3-8B	Gemma-2-9B
John	$11.004_{\pm 0.131}$	$9.909_{\pm 0.068}$	$11.432_{\pm 0.114}$
Mary	$10.438_{\pm 0.157}$	$9.501_{\pm 0.091}$	$11.471_{\pm 0.134}$
Tom	$11.072_{\pm 0.129}$	$9.354_{\pm 0.069}$	$10.469_{\pm 0.128}$
James	$8.700_{\pm 0.138}$	$8.578_{\pm 0.083}$	$9.944_{\pm 0.126}$
Dan	$8.637_{\pm 0.139}$	$7.208_{\pm 0.086}$	$8.585_{\pm 0.136}$
Sid	$5.426_{\pm 0.174}$	$5.675_{\pm 0.128}$	$6.253_{\pm 0.177}$
Martin	$7.373_{\pm 0.142}$	$7.359_{\pm 0.092}$	$8.502_{\pm 0.147}$
Amy	$8.158_{\pm 0.149}$	$8.169_{\pm 0.101}$	$9.664_{\pm 0.139}$
Alice	$9.915_{\pm 0.156}$	$9.165_{\pm 0.088}$	$10.784_{\pm 0.120}$
Bob	$10.093_{\pm 0.149}$	$8.973_{\pm 0.076}$	$10.740_{\pm 0.128}$
Charlie	$9.815_{\pm 0.160}$	$8.483_{\pm 0.098}$	$9.857_{\pm 0.144}$
Eve	$9.639_{\pm 0.135}$	$7.314_{\pm 0.120}$	$7.933_{\pm 0.156}$
Semantically Irrelevant Tokens			
sun	$1.931_{\pm 0.081}$	$2.545_{\pm 0.057}$	$3.537_{\pm 0.072}$
river	$2.025_{\pm 0.081}$	$1.974_{\pm 0.047}$	$2.838_{\pm 0.064}$
mountain	$0.701_{\pm 0.081}$	$0.672_{\pm 0.045}$	$2.305_{\pm 0.065}$
cloud	$1.245_{\pm 0.083}$	$1.280_{\pm 0.042}$	$2.916_{\pm 0.067}$
wind	$1.411_{\pm 0.081}$	$1.766_{\pm 0.048}$	$3.899_{\pm 0.073}$
rain	$1.719_{\pm 0.081}$	$1.732_{\pm 0.043}$	$4.092_{\pm 0.079}$
tree	$2.762_{\pm 0.092}$	$2.269_{\pm 0.050}$	$3.186_{\pm 0.071}$
stone	$1.155_{\pm 0.079}$	$1.758_{\pm 0.041}$	$2.618_{\pm 0.057}$
leaf	$0.897_{\pm 0.077}$	$1.509_{\pm 0.047}$	$1.755_{\pm 0.061}$
fire	$2.285_{\pm 0.085}$	$2.846_{\pm 0.045}$	$4.836_{\pm 0.072}$

Table 5: Average logits over 500 IOI samples with standard errors for semantically relevant (in orange) and irrelevant first answer tokens. The differences between each pair of tokens from the two categories are significant, with variations less than two standard errors of the difference distributions.

```
  - Leave the final [A] field blank in each generated sample.
```

We refer to Wang et al. [44, Appendix E.] for the IOI templates to use in verbatim.

Notes on Prompting LLMs. We strip the final "[A]" from each IOI sample, allowing LLMs to complete the sequence. It's important to note that different models typically use different tokenizers. To ensure the models behave as expected, we first compute transition scores with unclamped models to verify that the expected "[A]" is predicted with reasonable probabilities.

We observe that Llama-2-7B tokenizes empty spaces individually and treats the name "Eve" as two tokens. Specifically, it tokenizes " Eve" into two tokens, $[382, 345]$, with an optional begin-of-sequence token 1. In this case, we use token 345 to compute probabilistic changes. Additionally, for other models, we strip all empty spaces from the generated IOI samples, while for Llama-2-7B, we retain one space to ensure it generates content tokens directly.

D.3 Details on Semantic Entropy Probes

We share the layer concatenation details in Table 6 and show the comparison on the performances of Sparse SEPs against SEPs in Table 7.

D.4 Details on Activation Clamping

We include the details of activation clamping on various models in Table 8.

D.5 Details on SAE Training and Feature Latents

Sparse Auto-Encoders (SAEs) are used to learn sparse and interpretable representations of high-dimensional data, such as the hidden states of LLMs. SAEs consist of an encoder and a decoder that project hidden states into a higher-dimensional sparse feature space and then reconstruct them. The

Table 6: Model properties and selected layers for concatenation in SEPs and Sparse SEPs for *semantic entropy* in (L)ong-form and (S)hort-form generation settings.

Model Name	**No. Layers**	**Hid. Dim.**	**Layers of SEPs/Sparse SEPs**
Llama-3-70B (L)	80	8192	[36, 37, 38, 39, 40]
Gemma-2-9B (S)	42	3584	[38, 39, 40, 41, 42]
Llama-3-8B (S)	32	4096	[23, 24, 25, 26, 27]
Llama-2-7B (S)	32	4096	[21, 22, 23, 24, 25]

Table 7: ΔAUROC (x100) of Sparse SEPs (S. SEPs) compared to SEPs. Avg ± standard error. (S)hort- and (L)ong-form generations. Probes are trained on a concatenated set of performant layers. See Appendix B for details.

Model	**In-distribution**	**Generalization**
	(S. SEP − SEP)	(S. SEP − SEP)
Llama-3-8B (S)	0.95 ± 1.06	2.55 ± 2.81
Gemma-2-9B (S)	-3.67 ± 1.10	5.19 ± 1.43
Llama-2-7B (S)	1.08 ± 0.54	-1.59 ± 8.50
Llama-3-70B (L)	-2.26 ± 4.60	1.59 ± 3.89

model is trained with a loss function that combines the reconstruction error with an l_1 regularization term to encourage sparsity, activating only a few features per input.

We train Sparse Auto-Encoders (SAEs) on the post-MLP residual stream of the 25th layer of the Llama-3-8B model, following the recommendation by Templeton et al. [39] that this stream is less affected by cross-layer superposition. Our SAE expands the model hidden states from the activation space into a higher-dimensional feature space with an expansion factor of 16. We specifically focus on the 25th layer of the model, as layer-wise SE probes indicate that this layer provides the best performance (Fig. 1), and activation clamping at this layer yields remarkable performance (Section 5). The training is performed on 500 million tokens from the OpenWebText corpus [25].

We use the training objective of a Gated SAE, which is shown to be a Pareto Improvement over SAEs in the current evaluation standard (e.g., MSE of decoded activation and feature sparsity) of SAEs [36, 7].

Interpretable Features from SAE. To assess interpretability, we use both the activation and logit lens approaches. We leverage the resources provided by Neuronpedia [24], a platform trusted and utilized by, for example, Lieberum et al. [23], to list the tokens whose occurrences most strongly activate each feature (i.e. **activation lens**) as well as the tokens whose logits are most affected by the activation of each feature (i.e. **logit lens**). Features activated by a sufficient number of tokens are assigned an interpretation auto-generated by GPT-4o-mini [34]. We refer to Bricken et al. [5], Bills et al. [4] for guidance on generating auto-interpreted features.

See Table 9 for the full set of interpreted features. Interestingly, the features identified with clamped activation values, if not within the top sparsity percent (though some are), are mostly unavailable for interpretation since they are activated by too few tokens or have too low activation values.

Table 8: Details on activation clamping (e.g., clamped activation (clp. act.) and clamped layer (clp. lyr.) used in Section 5 across models in both (S)hort-form and (L)ong-form generation settings.

Model Name	N_{SE} **(Total)**	**Clp. Activation**	**Clp. Lyr.**
Llama-3-70B (L)	$[3099, 4031, 6970]_{(8192)}$	$low \rightarrow high$: [-0.24, 0.82, 3.00] $high \rightarrow low$: [-10.36, -4.93, 0.29]	$72_{(\text{out of } 80)}$
Llama-2-7B (S)	$[363, 1415, 2298]_{(4096)}$	$low \rightarrow high$: [14.49, 5.56, 26.25] $high \rightarrow low$: [-4.82, -9.00, -3.66]	$25_{(\text{out of } 32)}$
Llama-3-8B (S)	$[788, 2978]_{(4096)}$	$low \rightarrow high$: [3.73, -3.37] $high \rightarrow low$: [-1.59, 0.99]	$25_{(\text{out of } 32)}$
Gemma-2-9B (S)	$[1279, 2558]_{(3584)}$	$low \rightarrow high$: [-36.13, 76.50] $high \rightarrow low$: [44.94, -15.60]	$30_{(\text{out of } 42)}$

SE Category	**Feature ID**	**Auto-Interpreted Feature**
High SE	8735	Phrases indicating possession or belonging
	10857	The word "actually" and its frequency in the text, highlighting its relevance in various contexts
	14001	The conjunction "and" to identify connections or additions in sentences
	17664	Identifiers or codes related to technology and security issues
	25705	Concepts related to political bias and trust in science (e.g., citations)
	29370	Numerical values or references to quantitative data
	34230/2471	Vertical bars indicating section breaks or special formatting markers in the text (beginning of text)
	40224	Punctuation marks
	41748	Occurrences of the word "the" (often as the first word in a sentence)
	43052	Instances of punctuation and numerical values
	49386	References to public officials or roles mentioned in discussions
	51252	Technical terms and references to digital tools or features in various contexts
Low SE	21347	Promotional content related to sales and events
	43682	References to lunar missions and related space technology
	43686	Themes related to the relationship between technology and personal touch in human interactions
	43688	References to significant intellectual contributions and social theories
	43689	Concepts related to heritage and identity
	43690	References to legal or administrative processes regarding township formation
	43695	Phrases indicating political statements or commentary

Table 9: Auto-interpreted features corresponding to N_{SE} neurons under the high or low SE categories, generated by GPT-4o-mini [34]. The default number of selected features is 21, calculated as $10^{-3.505} \times d_{\text{model}} \times m$, where $10^{-3.505}$ represents the L_0 feature sparsity, $d_{\text{model}} = 4096$, and $m = 16$. Only features with sufficient activating tokens or interpretable activations are listed. Refer to Appendix D.5 for further evaluation details.