Evaluating Sparse Autoencoders with Delta Selectivity

Pratik Doshi
University of California, Santa Cruz
prdoshi@ucsc.edu

Abstract

Sparse Autoencoders (SAEs) are widely used to interpret features encoded in the embeddings of transformer-based Language Models. Despite widespread adoption, there are limited techniques that provide a fine-grained evaluation for SAEs. This paper proposes delta selectivity, a metric that captures the performance of SAEs at feature-level granularity. Using this metric, it is possible to evaluate SAEs on a defined set of features relevant to their end-use. We implement delta selectivity on all the layers of Pythia's 160m and 410m models using 14 predefined features. We demonstrate the effectiveness of the metric and highlight cases where delta selectivity disagrees with reconstruction loss, which is traditionally used to evaluate SAEs. We open-source our code so that other researchers can augment their studies using the delta selectivity metric.

1 Introduction

Specialized evaluation tasks have been developed to assess Large Language Models (LLMs) on various attributes—such as coding, math problem-solving, common sense, morality, etc. When a specific task is defined, we rely on these targeted evaluations to select the most capable LLM. We do not have this luxury for Sparse Autoencoders (SAEs).

Typically, SAE evaluation involves analyzing the reconstruction-sparsity frontier using mean-squared error for reconstruction loss (or some variation) and L0/L1 for sparsity. Apart from this, confidence in SAEs comes mainly from the manual studies that use human raters/experts to assess the quality of the explanations generated by the SAEs. Recently, researchers have been using LLMs to automatically generate and validate natural language explanations. While this has been successful so far, we don't have any targeted techniques to evaluate SAEs on their ability to capture the features of interest.

SAEs are presently used for several purposes, including ones critical for AI safety, like unsafe query detection and prompt refusal. In this paper, we propose a metric called delta selectivity that can measure how well an SAE captures a given feature. This metric allows targeted evaluations and the development of a general evaluation suite containing a wide range of features that can be used for fine-grained SAE benchmarking. Fine-grained SAE benchmarking allows researchers to deeply understand the implications of SAE architectural choices. For instance, researchers can use delta selectivity to pinpoint the features that are better captured by adopting a different activation function. The analysis does not have to be restricted to the reconstruction-sparsity tradeoff.

The contributions of this paper are as follows:

- Proposes the delta selectivity metric, that can be used for fine-grained SAE evaluations.
- Uses a subset of previously adopted probing datasets [8] to demonstrate the metric and analyzes its relationship with traditional reconstruction loss.
- Publishes an open-source codebase that can be used to compute selectivity on a given set of features: https://github.com/Pratik-Doshi-99/delta-selectivity-saes

Preprint. Under review.

2 Related Work

2.1 Sparse Autoencoders

Language Models (LMs) are able to pick up millions of features observed in the training corpus and effectively compress them into a few thousand embedding dimensions available to them. This is famously called the Superposition Hypothesis. As a result of such compression, a single neuron must represent several (often unrelated) features. This idea is called Polysemanticity. Polysemanticity works because the features represented occur sparsely in the training corpus. For instance, consider a feature that represents programming languages and another that represents legal ethics. These features are largely unrelated, and it is unlikely that they will occur together in a token sequence. This allows LMs to encode both these features using the same neuron. The LM can distinguish between those concepts on the basis of other neurons that may encode concepts related to programming or the legal profession. In general, polysemanticity makes LM interpretability noisy.

Sparse Autoencoders (SAEs) have been widely adopted by the Mechanistic Interpretability community because they remedy Polysemanticity. [5] found that SAEs were able to extract relatively mono semantic features (a neuron represents a single feature). Subsequent research was done to improve the activation functions for SAEs. [7] introduced the Top-k SAE, and [10] proposed the JumpReLU activation function and a method to train SAEs using that activation function. In this paper, we use Top-k SAEs to evaluate the delta selectivity metric.

SAEs can be thought of as a high-dimensional linear layer (whose dimensions are in multiples of the dimensions of the model) trained to be sparse and reconstruct the activations of the model (harvested from a particular point in its forward pass). SAEs provide the necessary bandwidth (large dimensions) and incentive (sparsity regulation) for the model to express its features in a relatively monosemantic fashion.

Mathematically, it looks like:

$$\begin{aligned} \mathbf{f}(\mathbf{x}) &:= \sigma\left(\mathbf{W}_{\text{enc}}\mathbf{x} + \mathbf{b}_{\text{enc}}\right), \\ \hat{\mathbf{x}}(\mathbf{f}) &:= \mathbf{W}_{\text{dec}}\mathbf{f} + \mathbf{b}_{\text{dec}} \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the input activation, $\hat{\mathbf{x}}$ is the reconstruction of the input activation, and σ is the activation function. In the case of TopK SAEs, the activation function picks the K-largest latents, and sets the rest to 0.

2.2 Evaluating Sparse Autoencoders

SAEs are trained on two objectives: sparsity and reconstruction. Reconstruction incentivizes the SAE to capture more information (features) from the model's activations. Sparsity incentivizes the SAE to focus on the dominant features by becoming an information bottleneck. Both objectives are in tension with each other. SAEs are evaluated on the basis of the reconstruction-sparsity tradeoff associated with them. Following are the top metrics used to assess SAEs:

- *Reconstruction Loss*: It is computed as the mean-squared error between the input activations of the model and the reconstructed activations produced by the SAE. This metric is usually computed on a per-token basis and aggregated for the entire evaluation dataset. [10] uses a normalized variant obtained by dividing reconstruction loss with the reconstruction loss obtained by always predicting the dataset mean.
- *Downstream Loss or Delta LM Loss*: This metric involves replacing the LM's activations with those reconstructed by the SAE and evaluating the impact on the cross entropy loss or Kullback-Leibler (KL) divergence of the LM. Both [10] and [7] evaluate using this metric.
- *L0*: L0 is a standard method of evaluating the sparsity of SAEs. It is a count of the number of non-zero latents in the SAE ($\mathbf{f}(\mathbf{x})$ from (1)). [10] uses L0, both to evaluate and train the SAE. [7] does not use any sparsity metric because TopK by itself is a strong sparsity guarantee.
- *L1*: L1 is usually used to train SAEs because it provides a gradient signal. L0, by itself, cannot provide a gradient signal ([10] uses Straight Through Estimators for this purpose). L1 is computed as the sum of absolute values of the SAE latents ($\mathbf{f}(\mathbf{x})$ from (1)).

- *Probe Loss*: [7] fits 61 1D probes on the SAE latents to assess how well can the annotated features be predicted using the SAE latents. There is a strong assumption about the presence of the feature when using this metric. Delta selectivity is conceptually similar to probe loss but overcomes the inherent assumption.

Despite these metrics, manual interpretability studies are the most common way of evaluating SAEs. Recently, there has been a rise in automated interpretability, where researchers use LLMs to both generate natural language explanations and evaluate those explanations.

Delta selectivity provides a data-driven method to evaluate SAEs. It can help identify SAEs with poor reconstruction of the feature of interest and reduce the need for manual interpretability. It uses probing datasets (feature-annotated datasets) and evaluates the ability of the SAE to predict the feature in comparison to the model's ability to predict the same feature. By comparing it with the model, it is possible to overcome the assumptions of the probing loss metric. If a feature is not present in a particular layer, the model's activations would not be able to predict it either. In this way, a low delta selectivity almost exclusively means poor SAE reconstruction.

2.3 Probing

Probing is a mechanism to determine whether the inner representations of a deep learning model encode a given feature of interest. It was initially proposed by [1] and uses a classifier to predict an annotated feature, given the inner representation of the model. High classification accuracy is evidence of the presence of the feature in the model's representations. In this paper, we compare the probing accuracy of the SAE with that of the model. This tells us how much of the feature information was lost in the SAE.

[2] discusses the promises and shortcomings of probing classifiers. The author highlights the problem of memorization. This problem occurs when the classifier is powerful enough to memorize the (inner representation, feature label) map. It is able to achieve high accuracy due to such memorization, not because the inner representation contains the feature. [9] proposes the idea of selectivity. Originally proposed for NLP tasks, selectivity utilizes a control task to prevent memorization from corrupting the probe's accuracy. We explore this idea in more detail in the section on Delta Selectivity.

3 Delta Selectivity

3.1 Selectivity

Probes are supervised classifiers that predict the properties of data using the representations of a model. Such classifiers can achieve high accuracy either because

- the representations encode the properties of the data, or
- the classifier just learned the task at hand (problem of memorization).

[9] proposes the use of control tasks to solve this problem. A control task is another probe that is fit on randomized feature labels. Performance in the control task can be solely attributed to memorization (since the target labels are randomized). An ideal probe would have high task accuracy and low control task accuracy. Selectivity is the difference between the task accuracy and the control task accuracy.

For our purposes, we define selectivity in the following way:

$$\begin{gathered} x_{l,t} \in \mathbb{R}^d \\ g : \mathbb{R}^d \to \{0, 1\} \\ \hat{g} : \mathbb{R}^d \to \{0, 1\} \end{gathered}$$

$$Selectivity(x_{l,t}, y_{l,t}) = \text{Acc}(g(x_{l,t}), y_{l,t}) - \text{Acc}(\hat{g}(x_{l,t}), y'_{l,t}) \tag{2}$$

where l and t represent the layer and token position, $x_{l,t}$ is the activations/embeddings, $y_{l,t}$ is the binary feature, $y'_{l,t}$ is binary feature with the class labels randomized (class frequency maintained), g and $\hat{g}$ are the task and control probes respectively.

3.2 Delta Selectivity

Delta selectivity is the difference between the selectivity of the LM and that of the SAE. Formally:

$$DeltaSelectivity(x_l, \theta_j) = Selectivity(f(x_l), \theta_j) - Selectivity(x_l, \theta_j) \tag{3}$$

where l and t represent the layer and token position, $x_{l,t}$ is the activations/embeddings, $y_{l,t}$ is the binary feature, $y'_{l,t}$ is binary feature with the class labels randomized (class frequency maintained), g and $\hat{g}$ are the task and control probes respectively.

3.3 Delta Selectivity

Delta selectivity is the difference between the selectivity of the LM and that of the SAE. Formally:

$$DeltaSelectivity(x_l, \theta_j) = Selectivity(f(x_l), \theta_j) - Selectivity(x_l, \theta_j) \tag{4}$$

where $f(x_l)$ comes from (1), $Selectivity(f(x_l), \theta_j)$ is the selectivity of the SAE and $Selectivity(x_l, \theta_j)$ is the selectivity of the model, θ_j refers to the class labels for feature j. Notice how we ignore the t dimension. In practice, we aggregate multiple token positions and calculate delta selectivity as if it were a single token. This aggregation depends on the feature under consideration. For instance, features relating to code syntax occur only at certain token positions in the sequence. When computing delta selectivity, we have a tensor of shape *[samples, relevant_token_positions, embeddings]*, which we convert to *[samples * relevant_token_positions, embeddings]*. We adjust the class labels accordingly.

3.4 Evaluation Methodology

We demonstrate delta selectivity on all the layers of Pythia-160m and Pythia-410m Language Models [4] and contrast them with the traditional reconstruction loss. In doing so, we present a detailed analysis of delta selectivity across all the layers of the two models, using 14 binary probes across 5 datasets from [8]. The architectural details of the LMs and the SAEs used in this paper are summarized in Table 1. The binary probes and the datasets are summarized in Table 2.

The models, SAEs, and probing datasets must be compatible with each other. The selected SAEs must be trained on the activations of the selected models, and the probing datasets must be from the corpus used to train the SAEs. In our case, the models and SAEs are trained on the Pile [6], and the probing dataset is also a subset of the Pile. Given these requirements, the Pythia family of models was the best fit. We use the pretrained SAEs from the sparsify library [3] for computations. In the future, as and when pretrained SAEs are available, we will extend this analysis to the larger models of the Pythia family, like the 6.9B variant.

4 Results

Experiments-wide Correlation: We perform a comparative analysis to understand whether delta selectivity scores of the SAEs are correlated to traditional reconstruction loss. Across all layers of both models, and all datasets, the correlation is 5%. Figure 1 shows a scatter of the two metrics. There seems to be no predictable relationship between the two metrics. The absence of such a relationship is a preliminary indicator that the delta selectivity metric provides substantial information that we do not get from traditional reconstruction loss.

Figure 1 makes the case for delta selectivity. Let's focus on quadrants 1 (positive delta selectivity, high reconstruction loss) and 3 (low reconstruction loss, low delta selectivity). A majority of the plot lies in these two quadrants. This means that in a majority of cases, one of the following happens:

- We successfully capture features (positive delta selectivity), but reconstruction loss is also high.

Table 1: LMs and SAEs used in this paper

Model/SAE	Layers	Dimensions	Description
Pythia-160m	12	768	A decoder-only transformer, trained on the Pile [6]
Pythia-410m	24	1024	A decoder-only transformer, trained on the Pile [6]
SAE-Pythia-160m	12	32768	A series of SAEs trained on the activations of every layer of Pythia-160m. Each SAE is trained on 8.2 billion tokens from the Pile [6].
SAE-Pythia-410m	24	65536	A series of SAEs trained on the activations of every layer of Pythia-410m. Each SAE is trained on 8.2 billion tokens from the Pile [6].

- Reconstruction loss is low (which would imply high fidelity), but we still lose features (delta selectivity is negative)

In both these cases, merely relying on reconstruction loss is misleading.

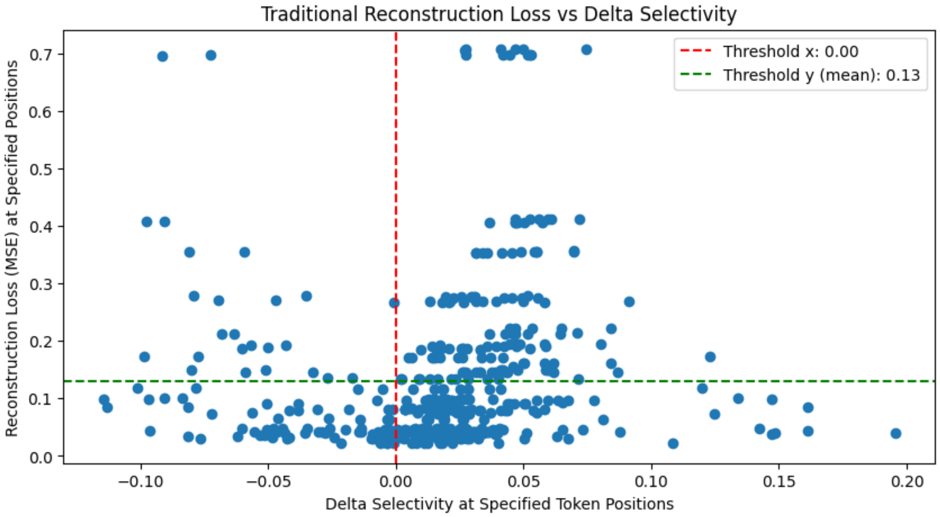

Figure 1: A scatter of reconstruction loss and delta selectivity. The plot confirms the absence of any predictable relationship.

Correlations by Model: We analyze the correlation between reconstruction loss and delta selectivity separately for both models. As shown in Figure 2, there is a stark difference between the two models. The correlations are stronger for the smaller variant, as compared to the larger variant. The distribution of delta selectivity doesn't vary between the two models as much as reconstruction loss does. The variation in the reconstruction loss explains why the correlations behave differently for different model sizes. A deeper analysis is required across multiple model sizes to understand the reliability of both metrics.

Table 3 contains the correlation between reconstruction loss and delta selectivity for each feature.

5 Conclusion

This paper proposes delta selectivity, a metric that captures the performance of SAEs in the wild. Using this metric, it is possible to evaluate SAEs on features that are of prime importance. We also show that this metric bears a low correlation with traditional measures of SAE fidelity, like

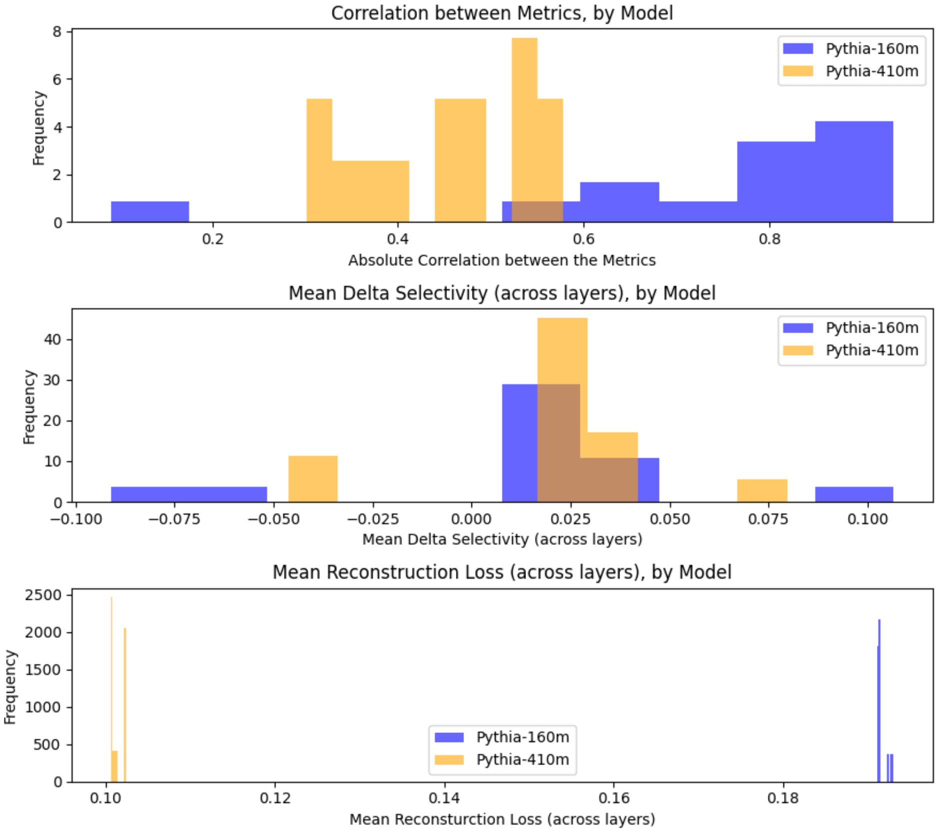

Figure 2: Analyzing the correlation between traditional reconstruction loss and delta selectivity, separately for both models.

reconstruction loss. It is, therefore, an important tool in the interpretability toolbox. We have quantitative metrics like cross entropy and test set accuracy to evaluate the performance of LLMs. Despite this, we resort to downstream evaluations because they are closer to the end-use and give a clearer picture of the suitability of LLMs for a given task. Delta selectivity plays a similar role for SAEs. At present, we are using SAEs for several AI safety cases, such as unsafe query detection or prompt refusal. Instead of relying on blanket metrics like reconstruction loss that don't give feature-specific results, delta selectivity can help evaluate which SAE can best capture the most important features for the task at hand.

References

[1] G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes, 2018.

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

[3] N. Belrose and L. Quirke. Sparsify: Sparsify transformers with saes and transcoders, 2024.

[4] S. Biderman, H. Schoelkopf, Q. G. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.

[5] T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. Turner, C. Anil, C. Denison, A. Askell, R. Lasenby, Y. Wu, S. Kravec, N. Schiefer, T. Maxwell, N. Joseph, Z. Hatfield-Dodds, A. Tamkin, K. Nguyen, B. McLean, J. E. Burke, T. Hume, S. Carter, T. Henighan, and C. Olah. Towards monosemanticity: Decomposing language models with dictionary learn-

ing. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[6] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[7] L. Gao, T. D. la Tour, H. Tillman, G. Goh, R. Troll, A. Radford, I. Sutskever, J. Leike, and J. Wu. Scaling and evaluating sparse autoencoders, 2024.

[8] W. Gurnee, N. Nanda, M. Pauly, K. Harvey, D. Troitskii, and D. Bertsimas. Finding neurons in a haystack: Case studies with sparse probing, 2023.

[9] J. Hewitt and P. Liang. Designing and interpreting probes with control tasks, 2019.

[10] S. Rajamanoharan, T. Lieberum, N. Sonnerat, A. Conmy, V. Varma, J. Kramár, and N. Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders, 2024.

Table 2: LMs and SAEs used in this paper

Feature	Dataset	Description
is_football	wikidata athlete	A dataset of text documents mentioning names of popular sports persons, probed at the names of those persons. The target class represents whether the person is a football player or not.
is_basketball	wikidata athlete	The target class represents whether the person is a basketball player or not.
is_baseball	wikidata athlete	The target class represents whether the person is a baseball player or not.
is_american_football	wikidata athlete	The target class represents whether the person is an American football player or not.
is_icehockey	wikidata athlete	The target class represents whether the person is an ice hockey player or not.
is_female	wikidata sex or gender	A dataset of text documents mentioning names of popular celebrities, probed at the names of those persons. The target class represents whether the person is a female (1) or male (0).
is_alive	wikidata is alive	A dataset of text documents mentioning names of popular celebrities, probed at the names of those persons. The target class represents whether the person is alive or not.
is_democratic	wikidata political party	A dataset of text documents mentioning names of popular political persons, probed at the names of those persons. The target class represents whether the person is a Democrat (1) or Republican (0).
is_singer	wikidata occupation	A dataset of text documents mentioning names of popular celebrities, probed at the names of those persons. The target class represents whether the person is a singer or not.
is_actor	wikidata occupation	The target class represents whether the person is an actor or not.
is_politician	wikidata occupation	The target class represents whether the person is a politician or not.
is_journalist	wikidata occupation	The target class represents whether the person is a journalist or not.
is_athlete	wikidata occupation	The target class represents whether the person is an athlete or not.
is_researcher	wikidata occupation	The target class represents whether the person is a researcher or not.

Table 3: Correlation between Delta Selectivity and Reconstruction Loss by Feature and Model

Feature	**Pythia-160m**	**Pythia-410m**
is_baseball	0.934187	0.335389
is_football	0.922726	0.540753
is_athlete	0.903483	0.471562
is_singer	0.887798	0.458328
is_american_football	0.859069	0.538522
is_democratic	-0.838297	-0.356850
is_basketball	0.827315	0.485343
is_icehockey	0.811831	0.455867
is_politician	0.768349	0.388122
is_journalist	0.684124	0.318750
is_researcher	0.611740	0.532410
is_actor	0.601520	0.574081
is_alive	0.544726	-0.300997
is_female	0.090856	-0.578412