Evaluating SAE interpretability without explanations

Gonçalo Paulo.
EleutherAI
goncalo@eleuther.ai

Nora Belrose
EleutherAI

Abstract

Sparse autoencoders (SAEs) and transcoders have become important tools for machine learning interpretability. However, measuring how interpretable they are remains challenging, with weak consensus about which benchmarks to use. Most evaluation procedures start by producing a single-sentence explanation for each latent. These explanations are then evaluated based on how well they enable an LLM to predict the activation of a latent in new contexts. This method makes it difficult to disentangle the explanation generation and evaluation process from the actual interpretability of the latents discovered. In this work, we adapt existing methods to assess the interpretability of sparse coders, with the advantage that they do not require generating natural language explanations as an intermediate step. This enables a more direct and potentially standardized assessment of interpretability. Furthermore, we compare the scores produced by our interpretability metrics with human evaluations across similar tasks and varying setups, offering suggestions for the community on improving the evaluation of these techniques.

1 Introduction

Sparse autoencoders (SAEs) and transcoders are now popular tools for interpreting large language models (Lieberum et al., 2024; Gao et al., 2024; Templeton et al., 2024), vision models (Surkov et al., 2024; Gorton, 2024) and other neural networks (Adams et al., 2025; Le et al., 2024). Naively interpreting the activations of neural networks tends to fail due to polysemanticity (Mu & Andreas, 2020; Zhang & Wang, 2023) — neurons usually fire in diverse, seemingly unrelated contexts. SAEs aim to overcome polysemanticity by representing each hidden state using an overcomplete basis, while requiring the coefficients of the linear combination to be sparse and non-negative. This allows SAE basis vectors (often called *latents*) to learn more specific and interpretable latents than those captured by neurons. Initial works showed that the latents of SAEs have very interpretable activations (Cunningham et al., 2023), specially in the highest quantiles. Recent work (Karvonen et al., 2025) has focused on generating good benchmarks that go beyond minimizing the reconstruction loss and increasing sparsity, instead focusing on possible downstream applications and addressing pathologies found in some SAEs, like feature absorption (Chanin et al., 2024).

The most popular methods for evaluating SAE interpretability involves generating a natural language explanation for each latent. This explanation is then evaluated by how useful it is for predicting the activation of the latent, or the effects of intervening on the latent, in a given context (Templeton et al., 2024; Paulo et al., 2024; Gur-Arieh et al., 2025). The use of natural language explanations is problematic, however. It complicates the evaluation pipeline, introducing additional hyperparameters and prompt choices which are likely to affect the final results. Ideally, we would like to abstract away from the details of explanation generation, focusing on the properties of the latents themselves.

The explanation-centered approach also assumes that an interpretable latent must have a meaning succinctly expressible in words (Ayonrinde et al., 2024; Paulo & Belrose, 2025). We question this philosophical assumption: we deem a latent interpretable if a human can *intuitively* understand its

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Mechanistic Interpretability Workshop.

meaning, and thereby distinguish activating from non-activating examples. In this work, we introduce two methods for assessing the interpretability of sparse coders, which do not require generating natural language explanations as an intermediate step. Our methods are applicable to both human and LLM evaluators, and we compare the scores produced by LLMs to those produced by humans.

2 Related work

2.1 Evaluating interpretability

We consider human evaluation to be the gold standard for interpretability, and LLM evaluators are valid only insofar as their judgments correlate strongly with those of humans. Chang et al. (2009) introduced the word intrusion task, where human participants identify an out-of-place word within a set of words associated with a specific topic discovered by a topic models. Subramanian et al. (2018) apply this approach to sparse word embeddings. The two alternative forced choice task (Borowski et al., 2020; Zimmermann et al., 2023) asks humans to discriminate between pairs of images where one strongly activates a feature map in a convolutional neural network, and one image does not. A model for which humans achieve high accuracy on these tasks, as well as a high inter-participant agreement rate, can then be considered interpretable.

Most methods of evaluating SAE interpretability involve generating an explanation for each latent, then evaluating the usefulness of these explanations for predicting whether, and/or how strongly, the latent will activate in a given context. This is similar to the simulation scoring method proposed in Bills et al. (2023) for neuron–based interpretability. However, this approach is entirely correlational, and may not capture the causal effects of latent activations on the model's output. Causal evaluations involve intervening on the model along a latent direction, and measuring how well the effect of this intervention can be predicted given a natural language explanation (Paulo et al., 2024; Gur-Arieh et al., 2025).

If SAE latents are strongly predictable by humans, we might imagine that we could replace the true latent activations from the SAE encoder with predicted activations from a human or LLM during a forward pass, and observe little performance degradation relative to the pristine model. This was attempted by Paulo & Belrose (2025), with mostly negative results. The idea is similar to concept bottleneck models (Koh et al., 2020), where the concepts are learned to explain an existing model, instead of learned as the model is trained.

2.2 Automatic explanation generation and evaluation

The standard method for generating explanations for latents involves collecting activations over a large corpus of text (Bills et al., 2023). Contexts that trigger the activation of a given latent are then presented to an LLM, which is tasked with summarizing them. Different strategies can be employed when sampling examples to show the LLM, such as stratified sampling from selected quantiles in the activation distribution (Templeton et al., 2024; Paulo et al., 2024).

Explanations can be evaluated via **simulation**, where an LLM is asked to predict a latent's activation strength at a specific token in a sentence given the explanation (Bills et al., 2023). However, this technique is computationally expensive and requires fairly heavyweight language models to produce meaningful results (Bills et al., 2023; Paulo et al., 2024). More recent methods aim to simplify this process by focusing on distinguishing between activating and non-activating examples (Paulo et al., 2024).

One such method is **detection**, where the model only needs to predict whether any token anywhere in an example activates a specified latent. In **fuzzing**, the model is asked whether a specific highlighted token in a context activates the latent or not. In Templeton et al. (2024), the authors propose a simple rubric score, where the model rates the relevance of a given interpretation in relation to the context and corresponding activations. Notably, these scoring techniques are more suitable for human evaluation due to their simplicity.

Other approaches focus on the downstream effects of activating certain latents. For instance, Gur-Arieh et al. (2025) and Paulo et al. (2024) generate explanations by analyzing the impact that a latent has on the base model's output when used as a steering vector. Their scoring metrics quantify how well the explanation does at predicting these steering effects.

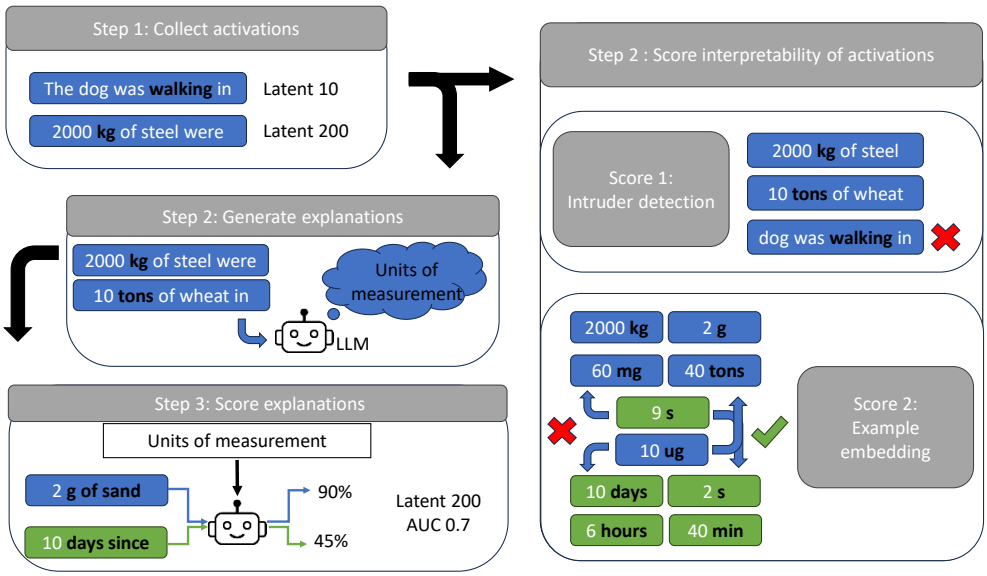

Figure 1: **Evaluating the interpretability of SAE latents.** To evaluate the interpretability of SAE latents, activations of the latents are collected over a colection of text. Traditionally these are then used to generate explanations, which are then scored. This process is an indirect measurement of the latents interpretability, and specific choices when generating explanations can influence the scores. Instead, we propose to evaluate the interpretability of SAE latents by directly looking at their activations, either through intruder detection or example embedding scoring.

3 Methods

In this work we focus on evaluating the interpretability of four different sparse autoencoders trained on the output of the MLPs at layers 9, 15, 21, and 27 of SmolLM2 135M (Allal et al., 2025). They were trained to minimize the mean squared error between their output and the MLP output, with no auxiliary loss terms. We adopt the state-of-the-art TopK activation function proposed by Gao et al. (2024), which directly enforces a desired sparsity level on the latent activations without the need to tune an L1 sparsity penalty, and consider the case where $k = 32$. All sparse coders are trained using the Adam optimizer (Kingma & Ba, 2017), a sequence length of 2049, and a batch size of 64 sequences. We train the SAEs on 10 billion tokens sampled from a reconstruction of the SmolLM2 training corpus.

3.1 Intruder detection

Inspired by intruder word detection (Chang et al., 2009; Klindt et al., 2025), we designed an intruder sentence detection task to evaluate the interpretability of SAE latents without relying on natural language explanations. For each latent, we sample four activating examples and one non-activating intruder example. The intruder is drawn from a pool of examples that do activate other latents but do not trigger the latent being evaluated. We randomly select one of the ten deciles of the activation distribution, then sample all of our activating examples from the same decile.

To evaluate the interpretability of each latent, we present the five examples to the LLM as a numbered list, and ask it to identify the intruder by index. Activating tokens in the activating examples are highlighted using $<<$ and $>>$ to provide additional context, while in non-activating examples, a random selection of tokens is highlighted. The number of highlighted tokens matches the average count in the activating examples, rounded down. Each example is constrained to 32 tokens in length to maintain consistency. Examples are preceded by a few-shot prompt demonstrating the task on synthetic examples.

The score of the latent is then defined as the accuracy of the evaluator at intruder detection for that latent, averaged over deciles. We also report the accuracy for individual deciles, allowing for the evaluation of interpretability as a function of activation strength.

We depart from traditional intruder word detection by showing a full context, because we find that a significant number of latents activate on more than one token in a context, sometimes on several adjacent tokens.

3.1.1 Human intruder detection

The authors manually performed the intruder task on a subset of examples to provide a human baseline. The procedure is similar to what is done by the LLM. The authors were shown five examples, with tokens highlighted as described above, and had to select which of the examples is the intruder. In LLMs, each intruder test is done in a fresh context, and so there is no information that carries from one set of examples to the next. The same is not true for the human participants. To minimize this effect, we randomly select which latent to sample the examples from, such that there aren't many consecutive examples of the same latent. The decile from which the examples are drawn are also chosen randomly and not shown to the participants.

3.2 Example embedding scoring

Although intruder detection is able to skip the explanation generation process, we find that a powerful LLM is still needed to produce meaningful scores. Since high-quality embedding models can be quite small, we explore a method we call example embedding scoring to speed up the evaluation process. It is inspired by the two alternative forced choice task used in Borowski et al. (2020); Zimmermann et al. (2024).

Example embedding scoring evaluates the interpretability of a latent by measuring whether activating examples cluster in embedding space. We sample sentence embeddings for a set of activating examples $E^+ = \{\mathbf{e}_1^+, \dots \mathbf{e}_n^+\}$ and a set of non-activating examples $E^- = \{\mathbf{e}_1^-, \dots \mathbf{e}_n^-\}$. We also sample embeddings for an activating example $\mathbf{q}_+$ called the **positive query**, and a non-activating example $\mathbf{q}_-$ called the **negative query**. We compute how much closer each query is to examples from its own class than to examples from the other class:

$$\Delta_+ = \frac{1}{N}\Big(\sum_{\mathbf{e}_i^+ \in E^+} \frac{\mathbf{q}^+ \cdot \mathbf{e}_i^+}{\|\mathbf{q}^+\|\|\mathbf{e}_i^+\|} - \sum_{\mathbf{e}_i^- \in E^-} \frac{\mathbf{q}^+ \cdot \mathbf{e}_i^-}{\|\mathbf{q}^+\|\|\mathbf{e}_i^-\|} \Big) \tag{1}$$

$$\Delta_- = \frac{1}{N}\Big(\sum_{\mathbf{e}_i^- \in E^-} \frac{\mathbf{q}^- \cdot \mathbf{e}_i^-}{\|\mathbf{q}^-\|\|\mathbf{e}_i^-\|} - \sum_{\mathbf{e}_i^+ \in E^+} \frac{\mathbf{q}^- \cdot \mathbf{e}_i^+}{\|\mathbf{q}^-\|\|\mathbf{e}_i^+\|} \Big) \tag{2}$$

As in intruder detection, all positive examples are sampled from the same decile of the activation distribution, and the negative examples are random non-activating contexts. We use the 22M parameter transformer all-MiniLM-L6-v2 as our embedding model.

Δ_+ and Δ_- can used to compute the AUROC, which is the scoring metric for this method. For each latent we iterate over different sets of explanations and queries, and average over them. This metric is similar to that of embedding scoring (Paulo et al., 2024), where explanations are substituted by activating examples.

4 Results

4.1 Intruder detection

The average intruder detection accuracy for a human is 65%, as measured on 56 different latents. On the highest activating deciles, human accuracy averaged 78%. In total, one third of latents had a score of over 80%, and only one in seven had a score lower than 30%, meaning that they few were considered non-interpretable. This supports the claim that explicit natural language interpretations are not needed to decide wether a given example should be active or not, and that showing other activating examples is sufficient.

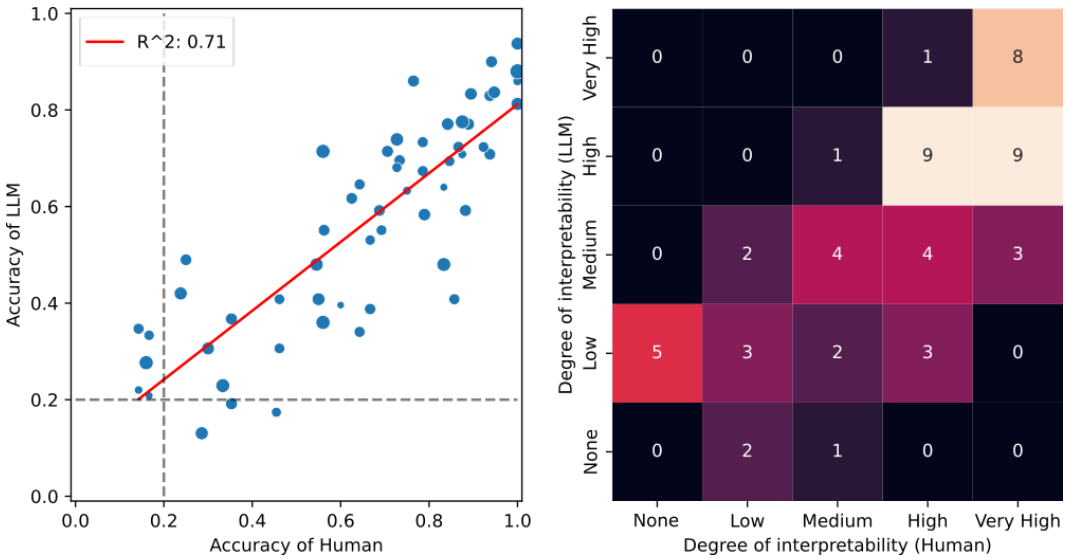

Figure 2: **Correlation between human and LLM intruder detection accuracy**. In the left panel we compare the accuracy on the intruder task for the LLM - in this case Claude Sonnet 3.5 - and that of a human. In the right panel we show a more coarse grained classification. All latents which have less than that 0.2 accuracy are considered non interpretable, and different degrees of interpretability are assigned to the other 4 bins of 0.2. The agreement between the "quality" of the latents given by humans and the LLM is significant, even though the LLMs underestimate the degree of interpretability. We evaluate 56 latents from SAEs trained on SmolLM 2 - around 14 latents per layer on four different layers. The size of each dot represents how many prompts the human saw for each latent, and is meant to represent the uncertainty associated with the score, with the smaller dots representing 8 prompts.

To validate the usage of LLMs instead of humans in the evaluation process, we compute the scores given by the LLM on the same latents. While humans only see on average 10 to 20 prompts per latent, Claude Sonnet 3.5, the LLM we compare to in the left panel of Figure 2, is evaluated on 50 prompts. The Spearman correlation between the LLM scores and the human score is 0.85, a higher agreement than what was seen in previous SAE evaluation metrics (Paulo et al., 2024), although in those works the human evaluators were not performing exactly the same task, as is the case here. The expected accuracy of random guessing is 20%, so each latent that has intruder detection accuracy of 20% or less is clearly uninterpretable. We create four additional interpretability bins, ranging from a low degree of interpretability (20-40%) to a very high degree of interpretability (80-100%). In the right panel of Figure 2 we report the agreement between binned human and binned LLM scores. We observe that the human mostly finds latents more interpretable than their LLM counterparts. This is good news, as it suggests LLMs are not finding convoluted patterns in the latent activations that humans cannot find.

Accuracy on the intruder task is dependent on the the activation decile from which the examples were sampled. Examples from the highest activating decile can have up to 20% higher accuracy than examples from the smallest activations (Figure 3 left panel).

Has found in previous work (Templeton et al., 2024; Paulo et al., 2024), examples with the lowest activations are hardest to interpret, even if we are considering only elements with similar activations. On the other hand, even though intruder detection accuracy is lower in the lower activating deciles, is still significantly higher than random. If we divide latents into interpretability bins, we see that the most interpretable latents remain interpretable even in the lower activation deciles, with accuracies greater than 0.75 (Figure A1), which we believe show that low activating examples can carry information about the behaviour of the latent.

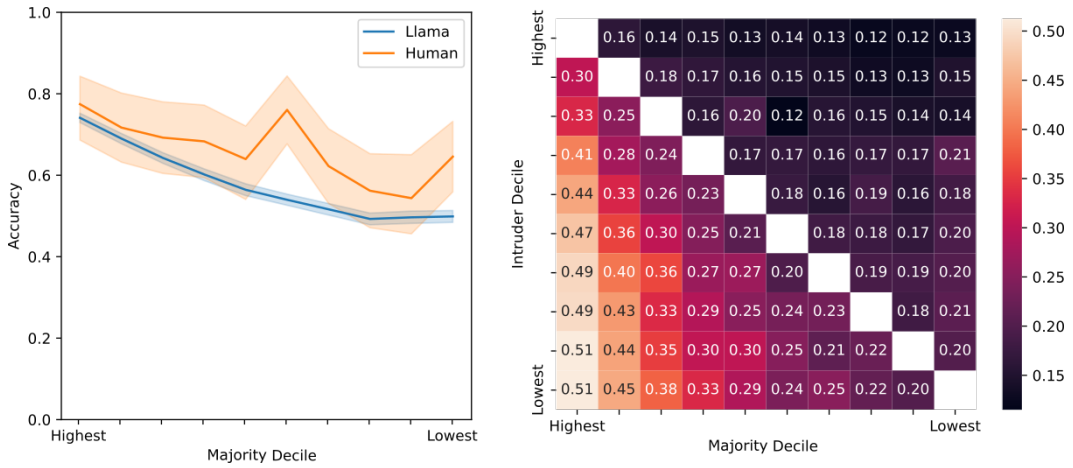

Figure 3: **Interpretability of the distribution of activations.** The activation examples used on the intruder task come from different deciles of the distribution of the latent's activations. On the lower activating deciles, it's harder to distinguish which is the intruder example, but the accuracy is still significantly above random. If instead of using non-activating examples we use examples from a different decile, we can observe that the accuracy of close deciles is close to random, while for further deciles it increases, although it is still lower than when using non activating examples. This shows that the full distribution of the activations of the SAE latents remains somewhat interpretable.

4.1.1 Detecting intruder deciles

Intruder detection makes it easy to investigate how the *meaning* of a latent varies as a function of the activation strength. Specifically, we modify our setup so that all five examples contain a token that activates the same latent, but one of them, the intruder, is sampled from a different activation decile as the rest. For a perfectly monosemantic latent representing a **binary** feature, we would expect the accuracy of this task to be close to random, as examples from different deciles would be very similar. If we assume that some latents represent **scalar** features with a notion of degree or intensity, one would expect that nearby deciles would exhibit close to random accuracy, and that distant pairs of deciles would be easier to distinguish. We would also expect symmetry, where the difficulty of finding a high-activating intruder among low-activating examples is similar to the accuracy of finding a low-activating intruder among high-activating examples.

In the right panel of Figure 3, we use Llama 3.1 70b (Team, 2024) to perform this detection task. As expected, Llama achieves highest accuracy when the majority decile is highest and the intruder decile is lowest (51%). This result is far from symmetric, however: when the majority decile is low, Llama essentially never achieves higher than random accuracy, no matter the decile of the intruder. Interestingly, when the intruder decile is high and the majority decile is low, we see significantly worse than random accuracy (around 13%).

While there likely are some 'perfect' scalar features in these SAEs, we do not see clear evidence for them in the right panel of Figure 3. The accuracy matrix is nonsymmetric, and even in the lower triangular it is not completely explained by the distance between majority and intruder deciles except by the highest activating deciles. We can rule out the hypothesis that most features are both binary and monosemantic, since we see far better than random accuracy for many (intruder, majority) pairs.

4.1.2 Inter-LLM and inter-human agreement

We investigate how the intruder accuracy for a given latent is correlated across different LLMs. We find that different models' scores have around the same correlation than with humans, except for the smaller Llama 3.1 8b, which has the worst accuracy (27%), and the lowest correlation with human judgment.

We are also interested in inter-human agreement. We find that, in a smaller set of 40 latents, each with only 3-5 prompts, the correlation between the two human labelers is of 0.87, while the one between

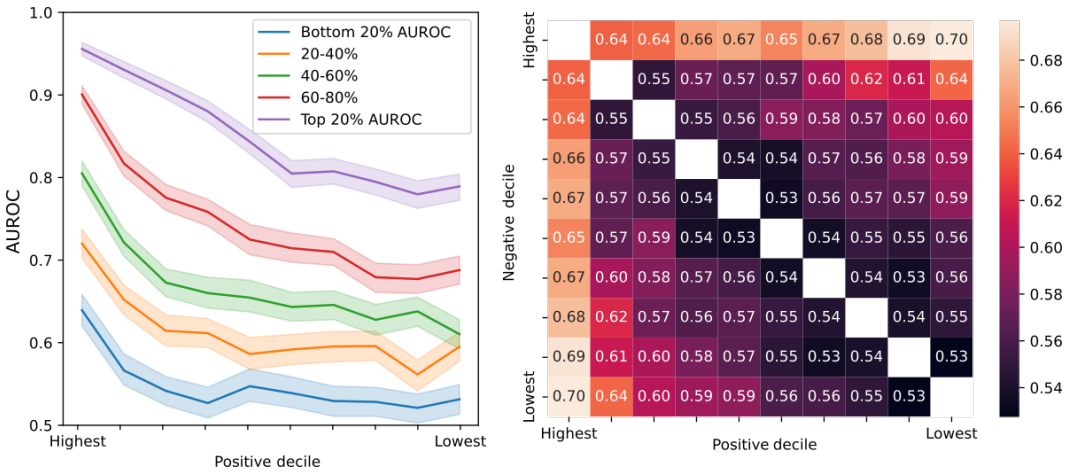

Figure 4: **Interpretability evaluation using example embedding scoring.** In the left panel, we show that the highest activating decile are easier to distinguish from non activating examples, as was the case with intruder detection. On the panel on the right, it is possible to see that, if using examples from different decile, we can observe that the accuracy of close decile is close to random, while for further decile it slightly increases. An exception to this is the first decile, which seems more different from the others.

Claude-Sonnet and each of the humans is 0.86 and 0.69. This level of inter-LLM and inter-human agreement is promising for this proposed method of evaluating SAE latent interpretability.

	Human	Llama 3.1 70b	Llama 3.1 8b	QwQ 32b	Gemini Flash 2.0	Claude Sonnet 3.5
Human	1	0.77	0.64	0.78	0.83	**0.84**
Llama 3.1 70b	0.77	1	0.58	**0.89**	0.85	0.88
Llama 3.1 8b	0.64	0.58	1	0.63	0.60	**0.65**
QwQ 32b	0.78	0.89	0.63	1	0.85	**0.91**
Gemini Flash 2.0	0.83	0.85	0.60	0.85	1	**0.87**
Claude Sonnet 3.5	0.84	0.88	0.65	**0.91**	0.87	1

Table 1: **Correlation between different evaluators.** We measure the Pearson correlation between the accuracy of intruder detection in 56 latents. Each latent was evaluated in 10-20 prompts by the human, and on 100 prompts by the LLMs.

4.2 Example embedding scoring

Example embedding scoring also finds that high deciles are easier than from low deciles (Figure 4 left panel). Unlike intruder detection accuracy, example embedding scoring is symmetric by construction (Figure 4 right panel). Unfortunately the observed AUROCs are fairly close to that of random guessing (0.5), even for deciles that are far apart. The highest decile is a notable exception, yielding AUROCs between 0.64 and 0.70.

We hypothesize that these results are due to the fact that activating tokens carry a lot of information about which decile an example belongs to, with the surrounding context being similar across deciles. The main weakness of embedding scoring is its low sensitivity to the activating token. Example embedding is a difficult task for an off-the-shelf embedding model, since it is unlikely to understand what our highlight tokens $<<$ and $>>$ mean.

Example embedding scores do not correlate as strongly with human intruder scores ($r = 0.48$) as LLM intruder scores do ($r = 0.84$). See Table 2 for full results. We suspect that example embedding scores tend to underestimate the interpretability of latents due to the small size of the embedding model and the difficulty of getting the model to attend to the tokens on which the latent is actually active, rather than the context as a whole. That said, it is a very fast and computationally efficient technique relative to other approaches.

	Example Embedding	**Intruder (LLM)**	**Intruder (Human)**	**Fuzzing F1**	**Detection F1**
Example Embedding	1	0.51	0.48	0.46	**0.58**
Intruder (LLM)	0.51	1	**0.84**	0.74	0.59
Intruder (Human)	0.48	**0.84**	1	0.69	0.50
Fuzzing F1	0.46	**0.74**	0.69	1	0.71
Detection F1	0.58	0.59	0.50	**0.71**	1

Table 2: **Correlation between different methods.**We measure the Pearson correlation between the accuracy of intruder detection and other scoring methods in 56 latents. Each latent was evaluated in 10-20 prompts by the human, and on 100 prompts by the other methods.

5 Conclusion

Prior work in the evaluation of SAEs relied on generating natural language explanations for SAE latents, which were then used to predict either the activations of those latents, or their causal effects. In this work we introduced two new methods for evaluating SAEs which do not rely on explanations: intruder detection, and example embedding scoring. We tested intruder detection using both human and LLM evaluators, finding that the human accuracies are highly correlated with those from LLMs. We use both intruder detection and example embedding scoring to measure the interpretability of different parts of the activation distribution, finding that higher deciles are more interpretable. As expected, example embedding scoring has a lower correlation with human intruder detection scores than LLM intruder detection scores do, but we believe it to be a promising method due to its speed: it uses a small embedding model rather than a large language model to perform the scoring.

References

Etowah Adams, Liam Bai, Minji Lee, Yiyang Yu, and Mohammed AlQuraishi. From mechanistic interpretability to mechanistic biology: Training, evaluating, and interpreting sparse autoencoders on protein language models. *bioRxiv*, pp. 2025–02, 2025.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Kola Ayonrinde, Michael T Pearce, and Lee Sharkey. Interpretability as compression: Reconsidering sae explanations of neural activations with mdl-saes. *arXiv preprint arXiv:2410.11179*, 2024.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*, 2, 2023.

Judy Borowski, Roland S. Zimmermann, Judith Schepers, Robert Geirhos, Thomas S. A. Wallis, Matthias Bethge, and Wieland Brendel. Exemplary natural images explain cnn activations better than state-of-the-art feature visualization, 2020. URL `https://arxiv.org/abs/2010.12606`.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-Graber, and David Blei. Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems*, 22, 2009.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders, 2024. URL `https://arxiv.org/abs/2409.14507`.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Liv Gorton. The missing curve detectors of inceptionv1: Applying sparse autoencoders to inceptionv1 early vision, 2024. URL https://arxiv.org/abs/2406.03662.

Yoav Gur-Arieh, Roy Mayan, Chen Agassy, Atticus Geiger, and Mor Geva. Enhancing automated interpretability with output-centric feature descriptions, 2025. URL https://arxiv.org/abs/2501.08319.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability, 2025. URL https://arxiv.org/abs/2503.09532.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

David Klindt, Charles O'Neill, Patrik Reizinger, Harald Maurer, and Nina Miolane. From superposition to sparse codes: interpretable representations in neural networks. *arXiv preprint arXiv:2503.01824*, 2025.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. *ArXiv*, abs/2007.04612, 2020. URL https://api.semanticscholar.org/CorpusID:220424448.

Nhat Minh Le, Ciyue Shen, Neel Patel, Chintan Shah, Darpan Sanghavi, Blake Martin, Alfred Eng, Daniel Shenker, Harshith Padigela, Raymond Biju, Syed Ashar Javed, Jennifer Hipp, John Abel, Harsha Pokkalla, Sean Grullon, and Dinkar Juyal. Learning biologically relevant features in a pathology foundation model using sparse autoencoders, 2024. URL https://arxiv.org/abs/2407.10785.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2, 2024. URL https://arxiv.org/abs/2408.05147.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *ArXiv*, abs/2006.14032, 2020. URL https://api.semanticscholar.org/CorpusID:220055965.

Gonçalo Paulo and Nora Belrose. Partially rewriting a transformer in natural language, 2025. URL https://arxiv.org/abs/2501.18838.

Gonçalo Paulo, Alex Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models, 2024. URL https://arxiv.org/abs/2410.13928.

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. Spine: Sparse interpretable neural embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Viacheslav Surkov, Chris Wendler, Mikhail Terekhov, Justin Deschenaux, Robert West, and Caglar Gulcehre. Unpacking sdxl turbo: Interpreting text-to-image models with sparse autoencoders, 2024. URL https://arxiv.org/abs/2410.22366.

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

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

Changwan Zhang and Yue Wang. A sample survey study of poly-semantic neurons in deep cnns. In *Conference on Computer Graphics, Artificial Intelligence, and Data Processing*, 2023. URL https://api.semanticscholar.org/CorpusID:258884786.

Roland S. Zimmermann, Thomas Klein, and Wieland Brendel. Scale alone does not improve mechanistic interpretability in vision models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=OZ7aImD4uQ.

Roland S. Zimmermann, David A. Klindt, and Wieland Brendel. Measuring mechanistic interpretability at scale without humans. In *ICLR 2024 Workshop on Representational Alignment*, 2024. URL https://openreview.net/forum?id=M8yBcvRvwn.

A Appendix

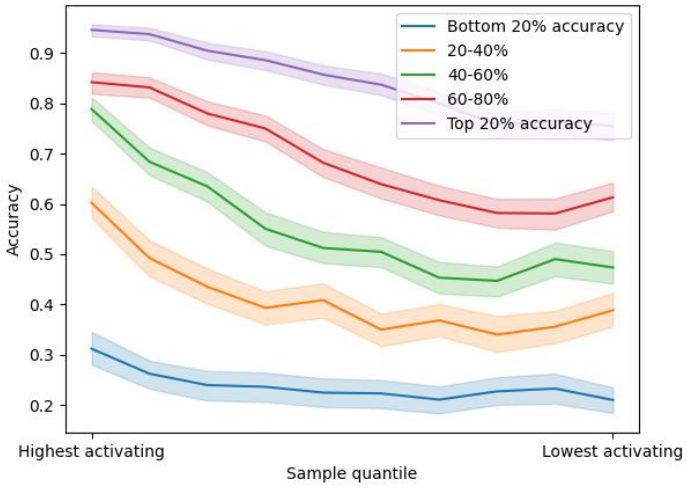

Figure A1: **Interpretability of the distribution of activations, stratified.** If we observe each of the 5 bins of accuracy separately, the bin with higher accuracy consistently has high accuracy throughout all the distribution, while the other bins have sharper drops in accuracy when going from the highest activating examples to the lowest ones.