A is for Absorption: Studying Feature Splitting and Absorption in Sparse Autoencoders

David Chanin[*,1,2] **James Wilken-Smith**[*,1] **Tomáš Dulka**[*,1] **Hardik Bhatnagar**[*,1,3]
Satvik Golechha[4] **Joseph Bloom**[1,5]

[1]LASR Labs [2]University College London [3]Tübingen AI Center, University of Tübingen
[4]MATS [5]Decode Research

Abstract

Sparse Autoencoders (SAEs) aim to decompose the activation space of large language models (LLMs) into human-interpretable latent directions or features. As we increase the number of features in the SAE, hierarchical features tend to split into finer features ("math" may split into "algebra", "geometry", etc.), a phenomenon referred to as feature splitting. However, we show that sparse decomposition and splitting of hierarchical features is not robust. Specifically, we show that seemingly monosemantic features fail to fire where they should, and instead get "absorbed" into their children features. We coin this phenomenon *feature absorption*, and show that it is caused by optimizing for sparsity in SAEs whenever the underlying features form a hierarchy. We introduce a metric to detect absorption in SAEs, and validate our findings empirically on hundreds of LLM SAEs. Our investigation suggests that varying SAE sizes or sparsity is insufficient to solve this issue. We discuss the implications of feature absorption in SAEs and some potential approaches to solve the fundamental theoretical issues before SAEs can be used for interpreting LLMs robustly and at scale.

1 Introduction

Large Language Models (LLMs) have achieved remarkable performance across a wide range of tasks, yet our understanding of their internal mechanisms lags behind their capabilities. This gap between performance and interpretability raises concerns about the "black box" nature of these models [30]. The field of mechanistic interpretability aims to address this issue by reverse-engineering the internal algorithms of neural networks and performing causal analysis on them [24].

Recent work theorizes that models represent concepts as linear directions in a high-dimensional space, known as the Linear Representation Hypothesis (LRH) [26, 8]. The model is able to represent far more concepts than it has neurons in its hidden space by allowing these concept directions to overlap slighly, known as superposition [8]. Superposition makes it challenging to directly interpret neurons in an LLM, and requires different techniques to extract interpretable feature directions.

As long as the active features in a given LLM activation are sparse and the underlying features follow the LRH, Sparse Autoencoders (SAEs) should be able to recover the true LLM features despite supersition using sparse dictionary learning [27]. Indeed, SAEs have shown potential in decomposing the dense, polysemantic activations of LLMs into more "interpretable" latent features [5, 2].

However, we show that even if all underlying features are linear and sparsely activating, an SAE will still fail to recover the true underlying features if the features form a hierarchy. Instead, an SAE will

[*]These authors contributed equally to this work.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Ideal interpretable solution

	SAE Encoder	SAE Decoder
Latent 1	"starts with S"	"starts with S"
Latent 2	"short"	"short"

The ideal interpretable solution requires firing **2** latents to represent "short".

Uninterpretable absorption solution

	SAE Encoder	SAE Decoder
Latent 1	¬"short" ∧ "starts S"	"starts with S"
Latent 2	"short"	"short" + "starts S"

Absorption only requires firing **1** latent to represent "short", and is unfortunately what the SAE learns.

Figure 1: In feature absorption, seemingly monosemantic latents fail to fire in cases where they apparently should. Here, we see an SAE can represent the word "short" and the concept "starts with S" more sparsely by absorbing the "starts with S" direction into the "short" latent, and then not firing the "starts with S" latent on the word "short", despite "short" starting with "S". Logical notation is used to describe the SAE encoder to emphasize its role as a classifier.

learn gerrymandered latents that fail to fire on seemingly arbitrary cases where the latent should fire according to the mainline interpretation of the latent. We refer to this failure as feature absorption.

Feature absorption is demonstrated in Figure 1, where the feature "short" always fires alongside a feature representing "starts with S". Instead of learning an interpretable latent representing "starts with S", the SAE can increase sparsity by instead disabling the "starts with S" latent when "short" is active while still getting perfect reconstruction.

We present the following contributions: (1) we identify a problematic variant of feature-splitting we call "feature absorption", where an SAE latent appears to track a human-interpretable concept, but fails to activate on seemingly arbitrary tokens. Instead, more specific latents activate and contribute a component of feature direction, "absorbing" the feature. (2) We demonstrate that feature absorption is caused by hierarchical features. (3) We develop a metric to detect feature absorption in LLM SAEs. And (4) we validate that feature absorption occurs in every LLM SAE we tested, including hundreds of open-source SAEs.

Feature absorption poses an obstacle to the practical application of SAEs since it suggests SAE latents may be inherently unreliable classifiers. This is particularly important for applications where we need confidence that latents are fully tracking behaviors, such as bias or deceptive behavior. Furthermore, techniques which seek to describe circuits in terms of a sparse combination of latents will also be more difficult in the presence of feature absorption [21].

An online explorer for our results can be found at `https://feature-absorption.streamlit.app`. Code is available at `https://github.com/lasr-spelling/sae-spelling`.

2 Background

Hierarchical features. We say features f_1 and f_2 form a hierarchy with f_1 as the parent and f_2 as the child if $f_2 \implies f_1$, meaning every time f_2 fires f_1 must also fire.

Linear probing. A linear probe is a simple linear classifier trained on the hidden activations of a neural network, typically using logistic regression (LR) [1].

K-sparse probing. A k-sparse probe [12] is a linear probe trained on a sparse subset of k neurons or SAE latents. Training a k-sparse probe first requires selecting the k best neurons or SAE latents that in-aggregate act as a good classifier, and then training a standard linear probe on just those k neurons or latents.

Gurnee et al. [12] proposed several methods of estimating the best k neurons or latents to pick, one of which involves first training a LR probe with a L1 loss term, and selecting the k largest elements by probe weight. When we refer to k-sparse probing in this work, we use this method of selecting k latents.

Sparse autoencoders. An SAE consists of an encoder, W_{enc}, a decoder, W_{dec}, and corresponding biases b_{enc} and b_{dec}. The SAE has a nonlinearity, σ, typically a ReLU (or variant such as JumpReLU [29]). Given input activation, a, the SAE computes a hidden representation, f, and reconstruction, $\hat{a}$:

$$f = \sigma(W_{enc}a + b_{enc}) \tag{1}$$
$$\hat{a} = W_{dec}f + b_{dec} \tag{2}$$

SAEs attempt to reconstruct input activations by projecting into an overcomplete basis using a sparsity-inducing loss term (typically $L1$ loss), or a certain number of non-zero latents ($L0$) on the hidden activations.

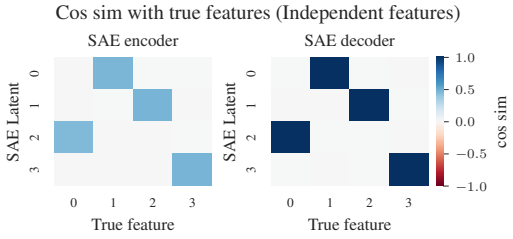
SAE feature ablation. In an ablation study we examine the downstream causal effect of an SAE latent by computing how patching its activation to 0 changes a downstream metric (e.g. logit difference). A negative ablation effect means intervening on the SAE latent would lower the metric. We follow the work of Marks et al. [21] and provide the algorithm in Appendix A.4.

3 Toy models of feature absorption

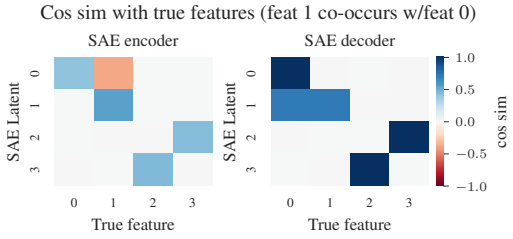
Absorption is caused by the SAE sparsity penalty in the presence of hierarchical features. When two features form a hierarchy, for instance "starts with S" and "short", the SAE can merge the "starts with S" feature direction into a latent tracking "short" and then not fire the main "starts with S" latent. This means firing one latent instead of two, increasing sparsity while retaining perfect reconstruction. We demonstrate that hierarchical feature co-occurrence causes absorption in a simple toy setting.

Our initial setup consists of 4 true features, each randomly initialized into orthogonal directions with a 50 dimensional representation vector and unit norm. Each feature fires with magnitude 1.0. Feature f_0 fires with probability 0.25, and features f_1, f_2, and f_3 fire with probability 0.05. Each SAE training input is created by sampling from these true features and summing the directions of each firing feature. We train a SAE with 4 latents to match the 4 true features using SAELens [15]. The SAE uses L1 loss with L1 coefficient 3e-5, and learning rate 3e-4. We train on 100M activations.

Independently firing features When the true features fire independently, we find that the SAE is able to perfectly recover these features as shown in Figure 2a. The SAE learns one latent per true feature. The decoder representations perfectly match the true feature representations, and the encoder learns to perfectly segment out each feature from the other features.



(a) When features fire independently, the SAE learns exactly one latent per feature, and the decoder perfectly recovers each feature direction.



(b) When features 0 and 1 co-occur (feature 1 only fires if feature 0 fires), we see absorption in the SAE encoder and decoder latents 0 and 1.

Figure 2: Comparison of independent features (left) vs. co-occurring features with absorption (right)

Hierarchical features cause absorption Next, we modify the firing pattern of feature 1 so it fires only if feature 0 also fires. We keep the overall firing rate of feature 1 the same as before, firing in 5% of activations. Features 2 and 3 remain independent.

Figure 2b shows the encoder and decoder cosine similarities with the true features in the hierarchical co-occurrence setup. Here, we see a clear example of feature absorption. Latent 0

Figure 3: Interpretation of learned SAE latents with co-occurrence between feature 0 and feature 1 (feature 1 only fires if feature 0 fires).

	SAE Encoder	SAE Decoder
Latent 0	$\neg$ feat 1 $\wedge$ feat 0	feat 0
Latent 1	feat 1	feat 0 + feat 1
Latent 2	feat 3	feat 3
Latent 3	feat 2	feat 2