Interpretability as Compression: Reconsidering SAE Explanations of Neural Activations

Anonymous Author(s)
Affiliation
Address
email

Abstract

Sparse Autoencoders (SAEs) have emerged as a useful tool for interpreting the internal representations of neural networks. However, naively optimising SAEs for reconstruction loss and sparsity results in a preference for SAEs that are extremely wide and sparse. We present an information-theoretic framework for interpreting SAEs as lossy compression algorithms for communicating explanations of neural activations. We appeal to the Minimal Description Length (MDL) principle to motivate explanations of activations which are both accurate and concise. We further argue that interpretable SAEs require an additional property, "independent additivity": features should be able to be understood separately. We demonstrate an example of applying our MDL-inspired framework by training SAEs on MNIST handwritten digits and find that SAE features representing significant line segments are optimal, as opposed to SAEs with features for memorised digits from the dataset or small digit fragments. We argue that using MDL rather than sparsity may avoid potential pitfalls with naively maximising sparsity such as undesirable feature splitting and that this framework naturally suggests new hierarchical SAE architectures which provide more concise explanations.

1 Introduction

Sparse Autoencoders (SAEs) (Le, 2013; Makhzani and Frey, 2013) were developed to learn a dictionary of sparsely activating features that describe a given dataset. They have recently become popular tools for interpreting the internal activations of large foundation language models, often finding human-understandable features (Sharkey et al., 2022; Huben et al., 2024; Bricken et al., 2023b).

Interpretability, in particular human-understandability, is difficult to optimise for since ratings—from humans or auto-interpretability methods (Bills et al., 2023)—are not differentiable at training time and often cannot be efficiently obtained. Researchers often use sparsity, the number of nonzero feature activations as measured by the L_0 norm, as a proxy for interpretability. SAEs are typically trained with an additional L_1 penalty in their loss function to promote sparsity.

We adopt an information theoretic view of SAEs, inspired by Grünwald (2007), which views SAEs as explanatory tools that compress neural activations into communicable explanations. This view suggests that sparsity may appear as a special case of a larger objective: minimising the description length of the explanations. This operationalises Occam's razor for selecting explanations: *all else equal, prefer the more concise explanation*.

We introduce this information theoretic view by describing how SAEs can be used in a communication protocol to transmit neural activations. We then argue that interpretability requires explanations to have the property of independent additivity, which allows individual features to be interpreted

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

separately and discuss SAE architectures that are compatible with this property. We find that sparsity (i.e. minimizing L_0) is a key component of minimizing description length but there are cases where sparsity and description length diverge. In these cases, minimizing description length directly gives more intuitive results. We demonstrate our approach empirically by finding the Minimal Description Length solution for SAEs trained on the MNIST dataset.

2 SAEs are communicable explanations

SAEs aim to provide explanations of neural activations in terms of "features"[1]. Here we reformulate SAEs as solving a communication problem: suppose that we would like to transmit the neural activations x to a friend with some tolerance ε, either in terms of the reconstruction error or change in the downstream cross-entropy loss. Using the SAE as an encoding mechanism, we can approximate the representation of the activations in two parts. ***First***, we send them the SAE encodings of the activations $z = Enc(x)$. ***Second***, we send them a decoder network $Dec(\cdot)$ that recompiles these activations back to (some close approximation of) the neural activations, $\widehat{x} = Dec(z)$.

This is closely analogous to *two-part coding schemes* (Grünwald, 2007) for transmitting a program via its source code and a compiler program that converts the source code into an executable format. Together the SAE activations and the decoder provide an **explanation** of the neural activations, based on the definition below.

Definition 2.1 *An explanation e of some phenomena p is a statement $e(p)$ for which knowing $e(p)$ gives some information about p. An explanation is typically a natural language statement*[2].

The description length (DL) of an explanation is the number of bits needed to transmit the explanation. For an SAE, this would be $DL = |z|_{\text{bits}} + |Dec(\cdot)|_{\text{bits}}$. The first term is O(n) and the second term is O(1) in the dataset size so the first term dominates in the large data regime.

Occam's Razor: All else equal, an explanation e_1 is preferred to explanation e_2 if $DL(e_1) < DL(e_2)$. Intuitively, the simpler explanation is the better one. We can operationalise this as the Minimal Description Length (MDL) Principle for model selection: Choose the model with the shortest description length which solves the task. It has been observed that lower description length models often generalise better (MacKay, 2003).

Definition 2.2 *We define the Minimal Description Length (MDL) as $MDL_{\varepsilon}(x) = \min DL(SAE)$ where $Loss(x, \widehat{x}) < \varepsilon$ and $\widehat{x} = SAE(x)$. We say an SAE is ε-MDL-optimal if it obtains this minimum.*

3 Interpretability requires independent additivity

Following Occam's razor we prefer simpler explanations, as measured by description length. But SAEs are not intended to simply give compressed explanations. They are also intended to give explanations that are interpretable and ideally human-understandable.

SAE features can be interpreted either as **causal results** of the model inputs (which we can see by analyzing feature activation patterns) or they can be interpreted as **causes** of the model outputs (which we can see through conducting interventions on the features and seeing the downstream effects). In both cases, we want to be able to understand each SAE feature independently, without needing to control for the activations of the other features. If all the feature activations are causally entangled—as is the case for the dense neural activations themselves—then they are not interpretable. Note that for D features there are $O(D^2)$ pairs of features and $\sum_i^K \binom{D}{i}$ possible sets of features

[1]Here we use the term "feature" as is common in the literature to refer to a linear direction which corresponds to a member of the set of a (typically overcomplete) basis for the activation space. Ideally the features are relatively monosemantic and correspond to a single (causally relevant) concept. We make no guarantees that the features found by an SAE are the "true" generating factors of the system.

[2]We will treat SAE activations and feature vectors as explanations themselves. Technically, we would want to do the additional step of interpreting their activation patterns or the results of causal interventions to get a natural language statement.

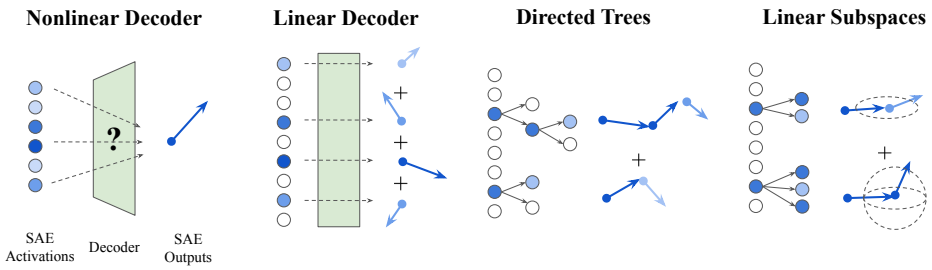

Figure 1: Examples of different SAE architectures. All but nonlinear decoders are compatible with independent additivity as feature activations correspond to adding a separate vector to the output. Architectures with directed tree decoders or which allow for vectors lying within a subspace are potentially more communication efficient since a child node can only be active if its parent node is active.

which is much too large for humans to hold in working memory. So for feature explanations to be human-understandable we cannot have the all the features being entangled such that understanding a single concept requires understanding arbitrary feature interactions.

Hence, for interpretability, we need to be able to understand features independently of each other such that understanding a collection of features together is equivalent to understanding all the features separately. We call this property **independent additivity**, defined below.

Definition 3.1 *Independent Additivity: An explanation e based on a vector of feature activations $\vec{z} = \sum_i \vec{z_i}$ is independently additive if $e(\vec{z}) \approx \sum_i e(\vec{z_i})$. We say that a set of features z_i are independently additive if they can be understood independently of each other and the explanation of the sum of the features is the sum of the explanations of the features*[3].

The independent additivity condition is directly analogous to the "composition as addition" property of the Linear Representation Hypothesis (LRH) discussed in Olah (2024). *Independent additivity* relates to the SAE features being composable via addition with respect to the explanation - this is a property of the SAE Decoder. In the Linear Representation Hypothesis however, *Composition as Addition* is about the underlying true features (i.e. the generating factors of the underlying distribution), which is a property of the underlying distribution.

It is immediate from the definition that Independent Additivity holds for linear decoders however, we note that this condition also allows for more general decoder architectures. For example, features can be arranged to form a collection of directed trees, shown in fig. 1, where arrows represent the property "the child node can only be active if the parent node is active"[4]. Here each feature still corresponds to its own vector direction in the decoder. Since each child feature has a single path to its root feature, there are no interactions to disentangle and the independent additivity property still holds, in that each *tree* can be understood independently in a way that's natural for humans to understand, as a multi-dimensional feature. An advantage of the directed-tree SAE decoder structure is that it can be more description-length efficient as shown in fig. 5.

Independent additivity of feature explanations also implies that the description length of the set of activations, $\{z_i\}$, is the sum of the lengths for each feature $DL(\{z_i\}) = \sum_i DL(z_i)$. If we know the distribution of the activations, $p_i(z)$, then it is possible to send the activations using an average description length equal to the distribution's entropy, $DL(z_i) = H(p_i) \equiv \sum_{z \in Z} -p_i(z) \log_2 p_i(z)$. For directed trees, the average description length of a child feature would be the conditional entropy, $DL_{\text{child}}(z_i) = H(p_i|\text{ parent active})$, which accounts for the fact that $DL = 0$ when the parent is not

[3]Note that here the notion of summation depends on the explanation space. For natural language explanations, summation of adjectives is typically concatenation ("big" + "blue" + "bouncy" + "ball" = "The big blue bouncy ball"). For neural activations, summation is regular vector addition ($\widehat{x} = \text{Dec}(\vec{z}) = \sum_i \text{Dec}(z_i)$).

[4]In practice, we typically expect feature trees to be shallow structures which capture causal relationships between highly related features. A particularly interesting example of this structure is a group-sparse autoencoder where linear subspaces are densely activated together.

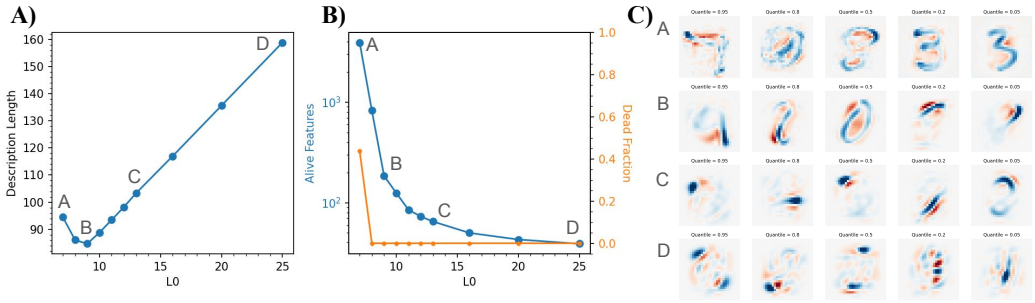

Figure 2: Finding the minimal description length (MDL) solution for SAEs trained on MNIST. A) Description length vs sparsity (L_0) for a set of hyperparameters with the same reconstruction error. B) Plot of the number of alive features as a function of sparsity (L_0). C) A random sample of SAE features at the 95th, 80th, 50th, 20th, and 5th percentiles of feature density respectively.

active. This is one reason that directed tree-style SAEs can potentially have smaller descriptions than conventional SAEs.

4 SAEs should be sparse, but not too sparse

Naively we might see SAEs as decompressing neural activations which contain densely packed features in superposition. To see that SAEs are producing compressed explanations of activations we must note that the inherent feature sparsity means that it is more efficient to communicate SAE latent features rather than neural activations even though the dimension of the latent dimension is higher.

The description length for a set of SAE activations (under independent additivity) with distribution $p(z)$ is given by $H(p) = \sum_{z \in Z} -p(z) \log_2 p(z)$. For exposition, consider a simpler formulation where we directly consider the bits needed without prior knowledge of the distributions. For a set of feature activations with L_0 nonzero elements out of D dictionary features, an upper bound on the description length is

$$DL \lesssim L_0(B + \log_2 D) \tag{1}$$

where B is the effective precision of each float and $\log_2 D$ is the number of bits required to specify which features are active. To achieve the same loss, higher sparsity (lower L_0) typically requires a larger dictionary, so there's an inherent trade-off between decreasing L0 and decreasing the dictionary size in order to reduce description length.

As an illustrative example, in Appendix B, we compare reasonable hyperparameters for GPT-2 SAEs to dense/narrow and sparse/wide extreme hyperparameters. We show that an SAE (Bloom, 2024) has a description length of approximately 1405 bits per input token, compared to 5376 bits for transmitting the dense neural activations and 13,993 bits for a one-hot encoding of all possible token sequences of length 128. Here the SAE at intermediate sparsity and width has the lower description length.

5 MDL-SAEs find interpretable and composable features for MNIST

Lee (2001) describe the classical method for using the Minimal Description Length (MDL) criteria for model selection. Here we choose between model hyperparameters (in particular the SAE width and expected L_0) for the optimal SAE. Our algorithm for finding the MDL-SAE solution and details for this case study are given in Appendix A.

We trained SAEs on the MNIST dataset of handwritten digits (LeCun et al., 1998) and find the set of hyperparameters resulting in the same test MSE. We see three basic regimes:

- **High L_0, narrow SAE width** (C, D in fig. 2): Here, the description length (DL) is linear with L_0, suggesting that the DL is dominated by the number of bits needed to represent the L_0 nonzero floats. The features appear as small sections of digits that could be relevant to many digits (C) or start to look like dense features that one might obtain from PCA (D).

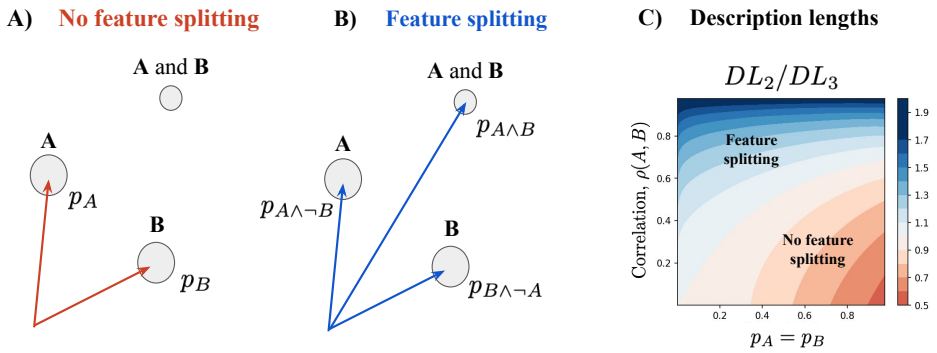

Figure 3: A toy model of undesirable feature splitting. The SAE can learn two boolean features without feature splitting (A) or three mutually exclusive boolean features with feature splitting (B) which always has lower L_0. Minimizing description length provides a decision boundary (C) for when feature splitting is preferred or not.

- **Low L_0, wide SAE width** (A in fig. 2): Though L_0 is small, the DL is large because as the SAE becomes wider, additional bits are required to specify which activations are nonzero. The features appear closer to being full digits, i.e. similar to samples from the dataset.
- **The MDL solution** (B in fig. 2): There's a balance between the two contributions to the description length. The features appear like longer line segments or strokes for digits, but could apply to multiple digits.

In this example, the MDL solution finds a meaningful decomposition of digits into stroke-like features. More dense SAEs find less interpretable point-like features, while sparser SAEs find features that resemble examples from the dataset and fail to decompose the digits into reusable and composable features.

6 Optimising for MDL can reduce undesirable feature splitting

In large language models, SAEs with larger dictionaries learn finer-grained versions of features learned in smaller SAEs, a phenomenon known as "feature splitting" (Bricken et al., 2023b). Feature splitting that introduces a novel conceptual distinction is desirable but some feature splitting—for example, learning dozens of features representing the letter "P" in different contexts (Bricken et al., 2023b)—is undesirable and can waste dictionary capacity while not giving more explanatory power.

A toy model of undesirable feature splitting is an SAE that represents the AND of two boolean features, A and B, as a third feature direction. The two booleans represent whether the feature vectors v_A and v_B are present or not, so there are four possible activations: 0, v_A, v_B, and $v_A + v_B$.

No Feature Splitting: Say that the SAE only learns two boolean feature vectors, v_A and v_B, as shown in fig. 3. It is still capable of reconstructing $A \wedge B$ as the sum $v_A + v_B$. The L_0 would simply be the expectation of the boolean activations, so $L_0 = p_A + p_B$ and the description length would be $DL = H(p_A) + H(p_B)$ where $H(p)$ is the entropy of a Bernoulli variable with probability p.

Feature Splitting: In this case, the SAE learns three mutually exclusive features. $A \wedge B$ is explicitly represented with the vector $v_A + v_B$ while the two other features represent $A \wedge \neg B$ and $B \wedge \neg A$ with vectors v_A and v_B. This setup has the same reconstruction error but has lower $L_0 = p_{A\wedge\neg B} + p_{B\wedge\neg A} + p_{A\wedge B} = p_A + p_B - p_{A\wedge B}$ since the probabilities for $A \wedge \neg B$, say, are reduced as $p_{A\wedge\neg B} = p_A - p_{A\wedge B}$. Note that the L_0 (sparsity) is necessarily lower than in the non-feature splitting case.

Even though feature splitting always results in a lower L_0, it does not always result in the smallest description length. The phase diagram in fig. 3 shows the case where $p_A = p_B$. If the correlation coefficient ρ between A and B is small then representing only A and B, but not $A \wedge B$, takes fewer bits so the preferred solution avoids feature splitting. However, if the correlation is large, then feature splitting is preferred since $A \wedge B$ occurs frequently enough that explicitly representing it reduces the description length. In this way, minimizing description length can limit the amount of undesirable

feature splitting and gives us a concrete decision criteria to understand when we might expect feature splitting.

7 Hierarchical features allow for more efficient coding schemes

Often features are semantically or causally related and this should allow for more efficient coding schemes. For example, consider the hierarchical concepts "Animal" (A) and "Bird" (B). Since all birds are animals, the "Animal" feature will always be active when the "Bird" feature is active. A conventional SAE would represent these as separate feature vectors, one for "Bird" (B) and one for "Generic Animal" ($A \wedge \neg B$), that are never active together, as shown in fig. 5. This setup has a low L_0, equal to the probability of "Animal", p_A, since something is a bird, a generic animal, or neither.

An alternative approach would be to define a variable length coding scheme (Salomon, 2007). For example, one might consider first sending the activation for "Animal" (A) and only if "Animal" is active, sending the activation for "Animal is a Bird" ($B|A$). Now the description length is given as $DL = H(p_A) + p_A H(p_{B|A})$ which is always fewer bits compared to the conventional SAE with $DL = H(p_A - p_B) + H(p_B)$, (see the phase diagram in fig. 5). The overall L_0 however is higher because sometimes two activations are nonzero at the same time, so $L_0 = p_A + p_{B|A}$.

This case illustrates the potential to reduce description length by matching the SAE architecture more closely to the hierarchical and causal structure of the data distribution. We also see another case where optimising for sparsity differs to the MDL approach - hierarchical structures of the type described above are never beneficial when optimising for sparsity but when thinking in terms of Description Length, there are clear benefits to using the semantic structure of the data.

8 Related Work

Bricken et al. (2023a) also consider how information measures relate to SAEs and find that "bounces" in entropy correspond to dictionary sizes with the correct number of features in synthetic experiments. We find a similar bounce in description length in a non-synthetic experiment. We go further by studying several examples where minimal description length gives more intuitive features and discuss more description-efficient SAE architectures.

As in Ramirez and Sapiro (2012), we use the MDL approach for the Model Selection Problem using the criteria that the best model for the data is the model that captures the most useful structure from the data. Chan et al. (2024) use Mechanistic Interpretability techniques to generate compact formal guarantees (i.e. proofs) of model performance and also note a deep connection between interpretability and compression.

9 Conclusion

In this work, we have presented an information-theoretic perspective on Sparse Autoencoders as explainers for neural network activations. Using the MDL principle, we provide some theoretical motivation for existing SAE architectures and hyperparameters. We also hypothesise a mechanism for, and criteria to describe, the commonly observed phenomena of feature splitting. In the cases where feature splitting can be seen as undesirable for downstream applications, we hope that, using this theoretical framework, the prevalence of undesirable feature splitting could be decreased in practical modelling settings.

Historically, evaluating SAEs for interpretability has been difficult without human interpretability ratings studies, which can be labour intensive and expensive. We propose that operationalising interpretability as efficient communication can help in creating principled evaluations for interpretability, requiring less subjective and expensive SAE metrics. We would be excited about future work which explores to what extent variants in SAE architectures can decrease the MDL of communicated latent feature activations. In particular, we suggest that exploiting causal structure inherent in the data distribution may be important to efficient coding.

References

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

J. Bloom. Open source sparse autoencoders for all residual stream layers of gpt2-small. *AI Alignment Forum*, 2024. URL https://www.alignmentforum.org/posts/f9EgfLSurAiqRJySD/open-source-sparse-autoencoders-for-all-residual-stream.

T. Bricken, J. Batson, A. Templeton, A. Jermyn, T. Henighan, and C. Olah. Features as the simplest factorization. *Transformer Circuits Thread*, 2023a. URL https://transformer-circuits.pub/2023/may-update/index.html#simple-factorization.

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

B. Bussmann, P. Leask, and N. Nanda. Batchtopk: A simple improvement for topk-saes. *AI Alignment Forum*, 2024. URL https://www.alignmentforum.org/posts/Nkx6yWZNbAsfvic98/batchtopk-a-simple-improvement-for-topk-saes.

L. Chan, R. Agrawal, A. Garriga-Alonso, and J. Gross. Compact proofs of model performance via mechanistic interpretability. *AI Alignment Forum*, 2024. URL https://www.alignmentforum.org/posts/bRsKimQcPTX3tNNJZ.

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. *arXiv preprint arXiv:2406.04093*, 2024.

P. D. Grünwald. *The minimum description length principle*. MIT press, 2007.

R. Huben, H. Cunningham, L. R. Smith, A. Ewart, and L. Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

Q. V. Le. Building high-level features using large scale unsupervised learning. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8595–8598. IEEE, 2013.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

T. C. Lee. An introduction to coding theory and the two-part minimum description length principle. *International statistical review*, 69(2):169–183, 2001.

D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

A. Makhzani and B. Frey. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013.

C. Olah. Circuits updates - july 2024, linear representations. *Transformer Circuits Thread*, 2024. https://transformer-circuits.pub/2024/july-update/index.htmllinear-representations.

I. Ramirez and G. Sapiro. An mdl framework for sparse coding and dictionary learning. *IEEE Transactions on Signal Processing*, 60(6):2913–2927, 2012.

D. Salomon. *Variable-length codes for data compression*. Springer Science & Business Media, 2007.

L. Sharkey, D. Braun, and B. Millidge. Taking features out of superposition with sparse autoencoders. *AI Alignment Forum*, 2022. URL https://www.alignmentforum.org/posts/z6QQJbtpkEAX3Aojj/interim-research-report-taking-features-out-of-superposition.

A SAE communication protocol

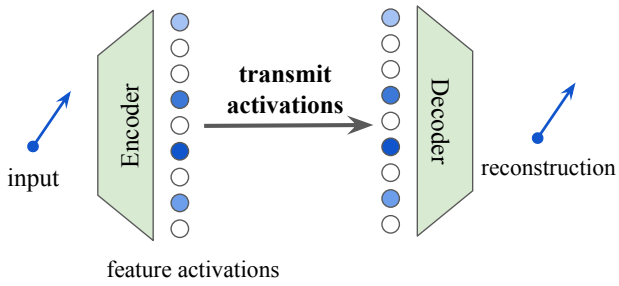

Figure 4: A schematic showing a sparse autoencoder (SAE) being used to communicate an input by transmitting the encoded activations and decoding them into a reconstruction of the input.

B Comparison of GPT-2 SAE hyperparameters

- **Reasonable SAEs**: Bloom (2024)'s open-source SAEs for GPT-2 layer 8 have $L_0 = 65$, $D = 25,000$. Given $B = 7$ bits per nonzero float (8-bit quantization with the sign fixed to positive), the description length per input token is 1405 bits.
- **Dense Activations**: A dense representation that still satisfies independent additivity would be to send the neural activations directly instead of training an SAE. GPT-2 has a model size of $d = 768$, the description length is simply DL = B d = 5376 bits per token.
- **One-hot encodings**: At the sparse extreme, our dictionary has a row for each neural activation in the dataset, so $L_0 = 1$ and $D = (\text{vocab size})^{\text{seq len}}$. GPT-2 has a vocab size of 50,257 and the SAEs are trained 128 token sequences. All together this gives $DL = 13,993$ bits per token.

Although the comparison is slightly unfair because the SAE is lossy (93% variance explained) and the other cases are lossless, these calculations demonstrate that reasonable SAEs are indeed compressed compared to the dense and sparse extremes. We hypothesise that the reason that we're able to get this helpful compression is that the true features from the generating process are themselves sparse.

Note the difference here from choosing models based on the reconstruction loss vs sparsity (L_0) Pareto frontier. When minimising L_0, we are encouraging decreasing L_0 and increasing D until $L_0 = 1$. Under the MDL model selection paradigm we are typically able to discount trivial solutions like a one-hot encoding of the input activations and other extremely sparse solutions which make the reconstruction algorithm analogous to a k-Nearest Neighbour classifier.

C Details on determining the MDL-SAE

C.1 Algorithm

1. **Specify a tolerance level, ε, for the loss function**. The tolerance ε is the maximum allowed value for the loss, either the reconstruction loss (MSE for the SAE) or the model's cross-entropy loss when intervening on the model to swap in the SAE reconstructions in place of the clean activations. For small datasets using a reconstruction, the test loss should be used.
2. **Train a set of SAEs within the loss tolerance**. It may be possible to simplify this task by allowing the sparsity parameter to also be learned.
3. **Find the effective precision needed for floats**. The description length depends on the float quantisation. We typically reduce the float precision until the change in loss results in the reconstruction tolerance level is exceeded.

4. **Calculate description lengths**. With the quantised latent activations, the entropy can be computed from the (discretized) probability distribution, $\{p_\alpha^i\}$, for each feature i, as

$$H = \sum_{i,\alpha} -p_\alpha^i \log p_\alpha^i$$

5. **Select the SAE that minimizes the description length** i.e. the ε-MDL-optimal SAE.

C.2 Details for MNIST case study

For MNIST, we trained BatchTopK SAEs (Bussmann et al., 2024), typically for 1000+ epochs until the test reconstruction loss converged or stopping early in cases of overfitting. Our desired MSE tolerance was 0.0150. Discretizing the floats to roughly 5 bits per nonzero float gave an average change in MSE of ≈ 0.0001, which was roughly the scale over which MSE varied for the hyperparameters used.

Gao et al. (2024) find that as the SAE width increases, there's a point where the number of dead features starts to rise. In our experiments, we noticed that this point seems to be at a similar point to where the description length starts to increase as well, although we did not test this systematically and this property may be somewhat dataset dependent.

D Description lengths for hierarchical features

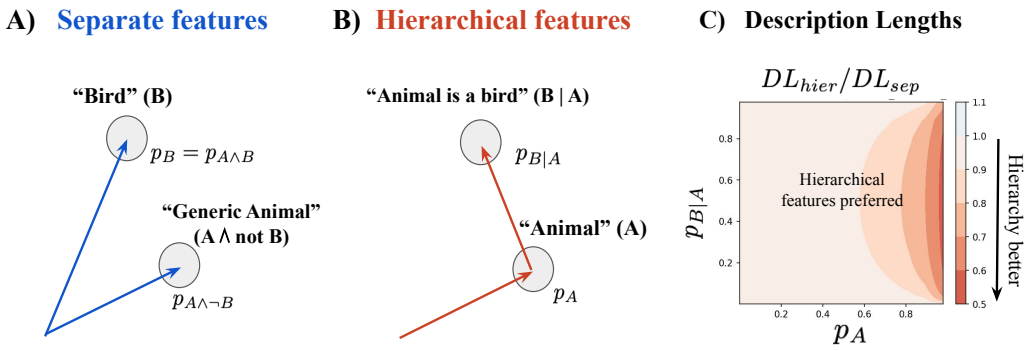

Figure 5: Two naturally hierarchical boolean features, such as "Animal" and "Bird", can be learned as separate mutually exclusive features (A) or in hierarchy (B) where the child feature can only be active if the parent feature is active, captured by the conditional probability $p_{B|A}$. C) The hierarchical case always has lower description length (DL) since the child feature's activations need not be sent when the parent is not active.