ADAPTIVE SPARSE ALLOCATION WITH MUTUAL CHOICE & FEATURE CHOICE SPARSE AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse autoencoders (SAEs) are a promising approach to extracting features from neural networks, enabling model interpretability as well as causal interventions on model internals. SAEs generate sparse feature representations using a sparsifying activation function that implicitly defines a set of token-feature matches. We frame the token-feature matching as a resource allocation problem constrained by a total sparsity upper bound. For example, TopK SAEs solve this allocation problem with the additional constraint that each token matches with at most k features. In TopK SAEs, the k active features per token constraint is the same across tokens, despite some tokens being more difficult to reconstruct than others. To address this limitation, we propose two novel SAE variants, *Feature Choice SAEs* and *Mutual Choice SAEs*, which each allow for a variable number of active features per token. Feature Choice SAEs solve the sparsity allocation problem under the additional constraint that each feature matches with at most m tokens. Mutual Choice SAEs solve the unrestricted allocation problem where the total sparsity budget can be allocated freely between tokens and features. Additionally, we introduce a new auxiliary loss function, `aux_zipf_loss`, which generalises the `aux_k_loss` to mitigate dead and underutilised features. Our methods result in SAEs with fewer dead features and improved reconstruction loss at equivalent sparsity levels as a result of the inherent adaptive computation. More accurate and scalable feature extraction methods provide a path towards better understanding and more precise control of foundation models.

1 INTRODUCTION

Understanding the internal mechanisms of neural networks is a core challenge in Mechanistic Interpretability. Increased mechanistic understanding of foundation models could provide model developers with tools to identify and debug undesirable model behaviour.

Dictionary learning with sparse autoencoders (SAEs) has recently emerged as a promising approach for extracting sparse, meaningful, and interpretable features from neural networks, particularly language models (Huben et al., 2024; Sharkey et al., 2022).

One problem with wide SAEs for foundation models is that there are often many dead features (Rajamanoharan et al., 2024a; Templeton et al., 2024; Gao et al., 2024). Dead features are features which are remain inactive across inputs, effectively wasting model capacity and hampering efficient training. Another problem is that approaches like TopK SAEs (Gao et al., 2024) don't have a natural way to take advantage of Adaptive Computation: spending more computation, and crucially more features, to reconstruct more difficult tokens.

We frame the problem of generating sparse feature activations corresponding to some given neural activations as a resource allocation problem, allocating the scarce total sparsity budget between token-feature matches to maximise the reconstruction accuracy. Within this framing, we naturally motivate two novel SAE variants which can provide Adaptive Computation: Feature Choice SAEs and Mutual Choice SAEs (FC and MC SAEs respectively). Feature Choice SAEs solve the sparsity allocation problem under the additional constraint that each feature matches with at most m tokens. Mutual Choice SAEs solve the unrestricted allocation problem where the total sparsity budget can be allocated freely between tokens and features. These approaches combine the Adaptive Computation of Standard SAEs with the simple optimisation and improved performance of TopK SAEs.

Our contributions are as follows:

- We provide a framing for sparsifying activation functions in SAEs as a solution to a resource allocation problem.

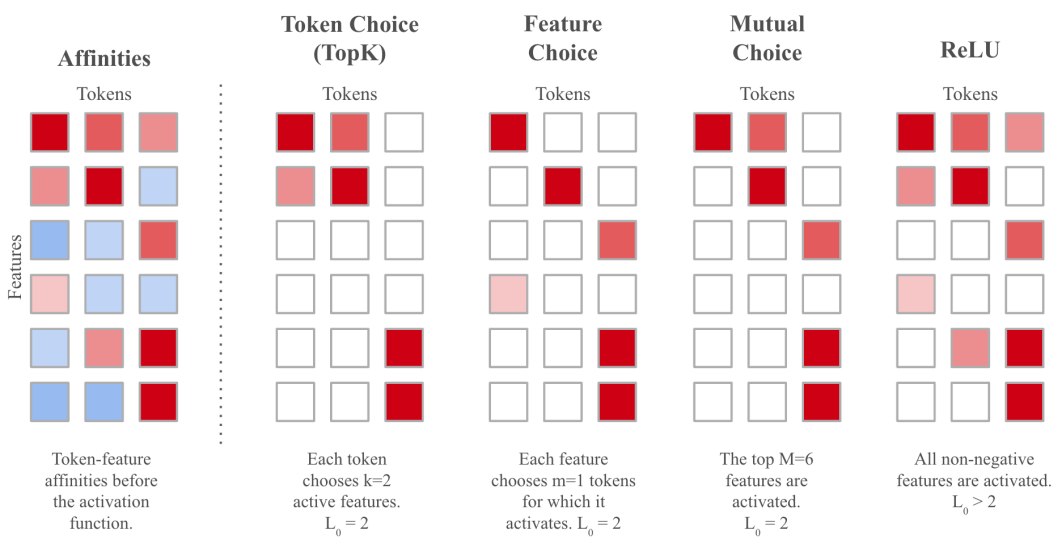

Figure 1: A comparison of the pre-activation affinities and the resulting feature activations following different sparsifying activation functions. Red and blue represent positive and negative affinities respectively, with deeper colours representing larger magnitudes. In the first three approaches we have a total sparsity budget of 6. *Affinities (Far-left)*: The token-feature affinities **Z'**, before any sparsifying activation function. *Token Choice/TopK (Center-left)*: We activate the top k features corresponding to each token. Note that there are features that don't fire in this batch, which could lead to dead features. *Feature Choice (Center)*: For each feature, it activates corresponding to the top m tokens with the highest affinity. Note that all features fire in this batch. *Mutual Choice (Center-right)*: The elements with the largest magnitude affinities activate, regardless of their token or feature affiliations. *ReLU/Standard (Far-right)*: All strictly positive elements activate. Here we allow low-magnitude feature activations which may be false positives and which cause the L_0 to be higher.

- We introduce two new SAE architectures: the **Mutual Choice SAE** and **Feature Choice SAE**, which are Pareto improvements on both standard SAEs and TopK SAEs. Additionally, the Feature Choice approach is, to our knowledge, the first SAE training method which reliably results in *zero dead features* even at large scale.
- We show that our methods naturally enable *Adaptive Computation*: using more features to reconstruct more difficult tokens. Instead of setting the number of features per token as a fixed k, we fix $\mathbb{E}[k]$ as a hyperparameter and allow the model to learn how to allocate the sparsity budget, without increasing computational overhead.
- We propose a novel auxiliary loss function, `aux_zipf_loss`, which mitigates under-utilisation of features and better utilises the SAE's full capacity.

We open-source a reference implementation for the community at [REDACTED]. We believe that with increasingly accurate approaches to feature extraction, it will become possible to connect sparse features over many layers and understand how models are computing outputs in a mechanistic, circuits-driven fashion. In particular, given that Feature Choice SAEs generally have no dead features, they can scale reliably to very large autoencoders, which are likely to be necessary for effective reconstruction on large foundation models such as GPT-4 (OpenAI et al., 2024) or Llama 3 (Dubey et al., 2024).

2 Related Work

2.1 Sparse Autoencoders

Sparse Autoencoders (SAEs) (Lee et al., 2007; Le, 2013; Mairal et al., 2014) learn an over-complete basis, or dictionary, of sparsely activating features. The feature activations, $\mathbf{z}$, correspond to their associated neural activations, $\mathbf{x}$, via the feature dictionary. In particular, we can write an SAE as:

$$\mathbf{z} = \sigma_s(\text{Enc}(\mathbf{x})) \in \mathbb{R}^F \tag{1}$$

$$\hat{\mathbf{x}} = \text{Dec}(\mathbf{z}) \in \mathbb{R}^N \tag{2}$$

where σ_s is a sparsifying activation function (e.g. ReLU), Dec is an affine map and $\mathbf{x} \in \mathbb{R}^N$ [1]. $\mathbf{z}' = \text{Enc}(\mathbf{x})$ are the pre-activation features, which we will call the token-feature *affinities*.

SAEs are trained to minimize the Reconstruction Error (Mean Squared Error) between $\mathbf{x}$ and $\hat{\mathbf{x}}$. This reconstruction error term is combined with an optional Sparsity Loss term (for example, an L_1 penalty to induce sparsity) and an optional Auxiliary Loss term to reduce dead features:

$$\mathcal{L}(\mathbf{x}) = |\mathbf{x} - \hat{\mathbf{x}}|_2^2 + \lambda_1 \mathcal{L}_{\text{sparsity}}(\mathbf{z}) + \lambda_2 \mathcal{L}_{\text{aux}}(\mathbf{x}, \mathbf{z}, \hat{\mathbf{x}}) \tag{3}$$

Rajamanoharan et al. (2024a); Templeton et al. (2024); Gao et al. (2024) have shown that decomposing neural activations using the SAE feature dictionary allows for increased human interpretability of models even at model sizes comparable to frontier foundation models.

2.2 Adaptive Computation

In **Adaptive Computation**, neural networks decide how much compute (and/or which parameters) to apply to a given input example (Graves, 2017; Xue et al., 2023). Ideally, the model should learn to apply less compute to easier examples and more compute to more difficult examples in order to maximise performance within a compute budget. In our setting, we consider the token-feature matches to be the scarce quantity to allocate, where we say that a token matches with a feature if the feature is activated on that token.

2.3 TopK SAEs

TopK SAEs (Gao et al., 2024) use a TopK activation function instead of the L_1 penalty to induce sparsity, as in Makhzani & Frey (2014). Though in the standard L_1 SAE formulation, the number of features-per-token is variable, in the TopK formulation the features-per-token is fixed at the same k for all tokens. We hypothesize that having a fixed k is a key drawback of the TopK method. Variable k values introduce Adaptive Computation which can focus more of the token-feature matching budget on more difficult tokens.

In concurrent work, Bussmann et al. (2024) introduce BatchTopK which is closely analogous to our Mutual Choice SAEs and also provides adaptive computation. However, they do not deal with the problem of underutilised features.

2.4 Dead Features

SAE features which remain inactive across many inputs are known as **dead features**. Bricken et al. (2023) declare a feature to be dead when it hasn't fired for at least 1e7 tokens. Dead features present a challenge especially when scaling to larger models and wider autoencoders. For example, Templeton et al. (2024) find 64.7% of features are dead for their autoencoders with 34M features. Our Feature Choice approach naturally results in *zero dead features* by ensuring that each feature activates for every batch.

2.5 Auxiliary k loss function

Gao et al. (2024) propose the auxiliary loss function, `aux_k_loss` to reduce the proportion of dead features. Given the SAE residual $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$, they define the auxiliary loss $\mathcal{L}_{\text{aux}} = |\mathbf{e} - \hat{\mathbf{e}}|^2$, where $\hat{e} = \text{Dec}(z_{\text{dead}})$ is the reconstruction using the top k_{aux} dead features. Gao et al. (2024) find fewer dead features when using the `aux_k_loss` for SAE training.

However, the `aux_k_loss` is only applied to features which qualify as dead. We apply a similar auxiliary loss, the `aux_zipf_loss`, to underutilised, but not yet dead, features.

3 Background

3.1 Sparsifying Activation Functions as Resource Allocators

We consider the following many-to-many matching problem:

- We have F features and a batch of B tokens. We would like to have at most M token-feature matches, where we say that a token matches with a feature if the feature is activated on that token [2] .

[1] In practice, there may be additional pre-processing and post-processing of the neural activations, $\mathbf{x}$.

[2] Here we follow the Mixture of Expert literature in abusing notation slightly to refer to neural activations corresponding to a given token position as a "token".

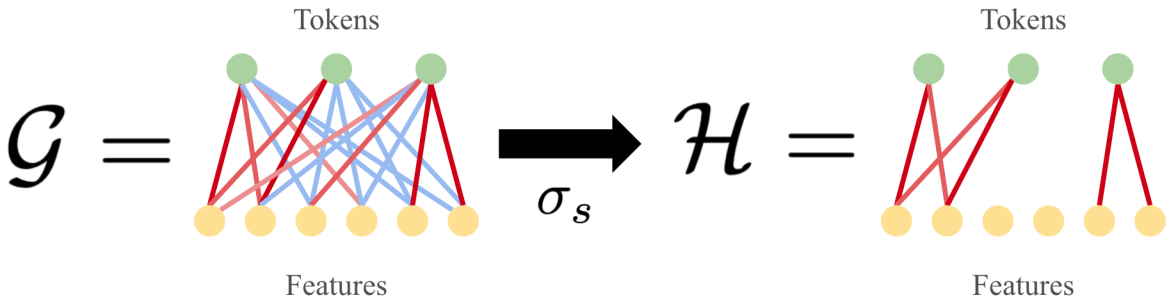

Figure 2: An illustration of the sparsifying activation function σ_s acting on the token-feature affinities. $\mathcal{G}$ (left) is a weighted bipartite graph $\mathcal{G} = \{\{T_1, T_2, T_3\} \times \{F_1, F_2, ..., F_6\}, \mathbf{E}\}$. Edge weights represent token-feature affinities, with red and blue representing positive and negative values respectively. We are seeking a subgraph $\mathcal{H} \subseteq \mathcal{G}$ with $M = 6$ edges. Here we have defined $\mathcal{H}$ (right) by the TopK method for $k = 2$; we select the 2 edges from each token with the largest edge weights. This provides an equivalent view to Figure 1 in terms of bipartite graphs.

- We would like to allocate our budget of M token-feature matches such that the reconstruction error is minimised.

Formally, we seek a reconstruction-error optimal weighted subgraph $\mathcal{H} \subseteq \mathcal{G} = \{\{1, ..., B\} \times \{1, ..., F\}, \mathbf{E}\}$ where $\mathcal{H}$ has at most M edges, for $M \ll BF$. Figure 2 details an example of a given graph $\mathcal{G}$ and subgraph $\mathcal{H}$. The edge weights can be viewed as the token-feature affinities: the pre-sparsifying activation feature magnitudes $\mathbf{z}'$. [3]

This problem doesn't immediately admit an efficient solution because it is currently unspecified how the edge weights contribute to the token reconstruction error. We make a simplifying assumption that we denote the **Monotonic Importance Heuristic** - the edges with the largest edge weights are likely to represent the most important contributions to the reconstruction error. With this heuristic, we can solve the problem of allocating token-feature matches by choosing the M edges with the largest magnitude edge weights as the edges for our subgraph $\mathcal{H}$.

We can equivalently view this allocation problem as choosing a binary mask $\mathbf{S} \in \mathbb{R}^{B \times F}$ with at most M non-zero elements which maximises reconstruction accuracy. This mask is to be element-wise multiplied with a token-feature affinity matrix $\mathbf{z}' \in \mathbb{R}^{B \times F}$. Applying the Monotonic Importance Heuristic, we are looking for the mask $\mathbf{S}$ such that $\sum_{i,j} \mathbf{z}'_{i,j} = \sum_{i,j} \mathbf{z}'_{i,j} \odot \mathbf{S}_{i,j}$ is maximised.

TopK SAEs: We can now see the TopK SAE approach as a special case of the above allocation problem, with the additional constraint that the number of features per token is at most k for each token. In other words, $\sum_i (\mathbf{S}_{i,j}) = k \; \forall j$, where $M = kB$. This leads to the solution of $\mathbf{S} = \text{TopKIndices}(\mathbf{z}', \text{dim} = -1)$, i.e. $\mathbf{S}$ picks out the k features with the highest affinity for each token [4]. Here $\sigma_s(\mathbf{z}') = \mathbf{S} \odot \mathbf{z}'$; element-wise multiplication with $\mathbf{S}$ defines our sparsifying activation function σ_s.

We now consider two other variants of this problem displayed in Figure 1:

Feature Choice SAEs: Whilst TopK SAEs require each token to match with at most k features, we instead add the constraint that each feature matches with at most m tokens. This is equivalently a constraint on the columns of S rather than its rows: $\sum_j (\mathbf{S}_{i,j}) = m \; \forall i$, where $M = mF$. This leads to the solution of $\mathbf{S} = \text{TopKIndices}(\mathbf{z}', \text{dim} = 0)$, i.e. $\mathbf{S}$ picks out the m tokens with the highest affinity for each feature.

Mutual Choice SAEs: Here we don't add any additional constraints and allow any choice of token-feature matching. This leads to the solution of $\mathbf{S} = \text{TopKIndices}(\mathbf{z}', \text{dim} = (0, 1))$, i.e. $\mathbf{S}$ picks out the largest elements of the $\mathbf{z}'$ affinity matrix, regardless of their position.

The Feature Choice (FC) and Mutual Choice (MC) sparsifying activation functions can be seen as having the desirable properties of the TopK activation (for example, preventing activation shrinkage, reducing the impact of noisy, low magnitude activations, allowing for a progressive recovery code, enabling simple model comparison, not requiring sparsity losses which are in conflict with the reconstruction loss etc.) but whilst allowing for Adaptive Computation, see

[3] As an illustrative example of the same problem, we can imagine a university which is able to confer at most M degrees where a student may take many degree subjects and a degree subject may admit many students.

[4] See Appendix H for a proof that this is the optimal solution under the constraints.

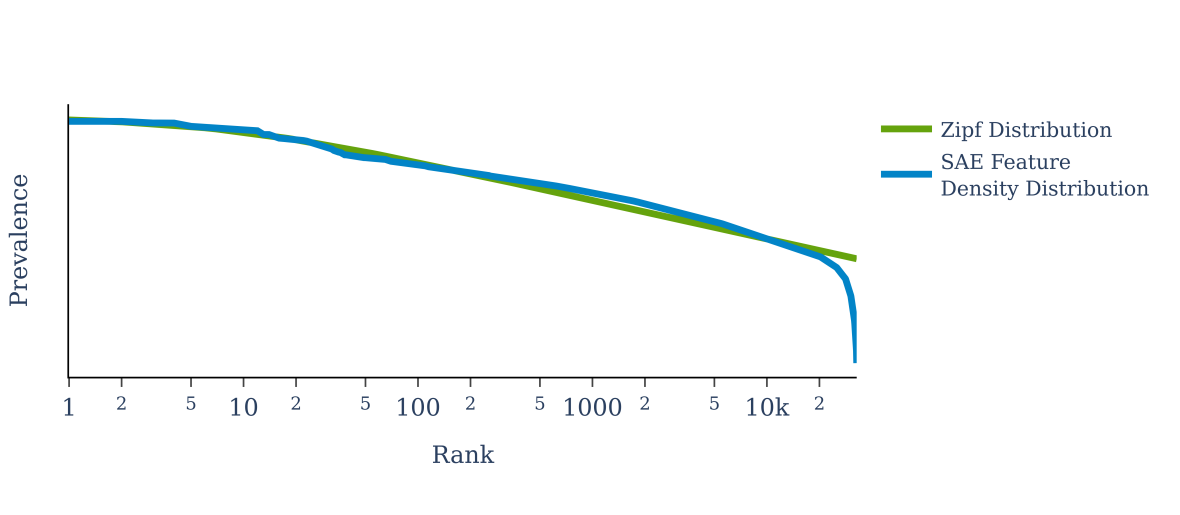

Figure 3: The Feature Density Curve fits a Zipf curve with $R^2 = 0.982$. The middle segment of the feature density distribution (features 100-20,000) fit the Zipf curve with R^2=0.996

Figure 1. Since we do not have constraints on the number of features per token in FC or MC, it's possible for one token to activate (i.e. match with) more features than another token.

3.1.1 Analogy To Mixture of Experts Routing

We choose our naming here to align with the study of token-expert matching in the Mixture of Experts paradigm, where there is a close analogy. Token Choice MoE routing strategies (Shazeer et al., 2017; Fedus et al., 2022) have the constraint that each token can be routed to at most k experts allowing for expert imbalances. On the other hand, Expert Choice routing strategies (Zhou et al., 2022) have the constraint that each expert processes at most m tokens, which eliminates the possibility of underutilised experts and allows tokens to be routed to a variable number of experts.

Here the intuitions are "*each token picks the k experts to be routed to*" and "*each expert picks the m tokens to be routed to that expert*" for Token Choice and Expert Choice respectively. The variety of MoE routing algorithms is explored in Liu et al. (2024). The *Feature Choice* approach we propose is directly analogous to the *Expert Choice* approach in MoEs and *TopK* SAEs are directly analogous to *Token Choice* MoEs. For this reason, we will call TopK SAEs, Token Choice SAEs to unify our notation.

3.2 Distribution of Feature Densities

When analysing the distribution of feature densities in open-source SAEs from Gao et al. (2024), we find (as shown in Figure 3) that the distributions typically follow a power law described by the Zipf distribution with R^2 = 0.982.

We note that although the Zipf distribution well fits most of the distribution, there is a considerable residual at the lower end of the distribution (the 20k+ rank tokens). We suggest that these features are underutilised. Underutilised features see fewer gradient updates than other features leading to a self-reinforcing cycle of these features being less useful and hence further underutilised until they die.

We will refer to these underutilised features as **dying features**. We define a dying feature as a feature which is one of the 25% least prevalent features which is also <60% as prevalent as we might predict from the prevalence rank of the feature using the fitted Zipf curve. More formally:

Definition 3.1. *Given a feature set $\mathcal{F} = \{\mathbf{f_1}, \mathbf{f_2}, ..., \mathbf{f_F}\}$ ordered by feature prevalence*[5]*, we say $\mathbf{f_i}$ is a **dying feature** if:*

1. $i > \frac{3}{4}F$ *- i.e. is in the bottom quartile of features when ranked by prevalence*
2. $\frac{\textit{actual prevalence}(\mathbf{f_i})}{\textit{expected prevalence}(\mathbf{f_i})} \equiv \frac{\textit{actual prevalence}(\mathbf{f_i})}{\textit{Zipf}(i)} < 0.6$

[5]That is feature $\mathbf{f_i}$ is the i'th most prevalent feature

3. $\forall j > i, \mathbf{f_j}$ *is also a dying feature*

Previous approaches to dealing with dead features either resampled dead features (Bricken et al., 2023) or applied gradients to dead features (for example `aux_k_loss` (Gao et al., 2024)) but they didn't address dying features. We hypothesise that many of the revived dead features were still not appropriately utilised.

4 METHODS

4.1 MUTUAL CHOICE AND FEATURE CHOICE ACTIVATION FUNCTION

As detailed in Section 3.1, we introduce two activation functions for SAEs: the *Mutual Choice and Feature Choice TopK activation functions*. Our SAEs have the same structure as standard (ReLU) SAEs and TopK SAEs except for in activation function:

- **Mutual Choice Activation Function**: $\mathbf{z} = \sigma_s(\mathbf{z}') = \text{TopK}(\mathbf{z}', k = M, \text{dim} = (0, 1))$;
- **Feature Choice Activation Function**: $\mathbf{z} = \sigma_s(\mathbf{z}') = \text{TopK}(\mathbf{z}', k = m_i, \text{dim} = 0)$;

4.2 REDUCING THE PROPORTION OF DYING FEATURES

To address the problem of dying features discussed in Section 3.2, we add an additional auxiliary loss for dying features, which is a natural generalisation of the `aux_k_loss`. Given the SAE residual $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$, we define the auxiliary loss $\mathcal{L}_{\text{aux\_zipf}} = |\mathbf{e} - \hat{\mathbf{e}}|^2$, where $\hat{\mathbf{e}} = \text{Dec}(\mathbf{z}_{\text{dying}})$ is the reconstruction using the top k_{aux} dying features. We can think of this `aux_zipf_loss` acting *preventatively* on features which could be at risk of becoming dead and acting *rehabilitatively* on features which have been recently revived. In this way, we reduce the proportion of both dead and dying features.

4.3 CHOOSING THE FEATURE CHOICE CONSTRAINT

In the Feature Choice approach, there remains the question of how to distribute the sparse feature activations across the feature dimension.

The simplest approach to this is to take $m_i = {}^{M}/_{F}$ for all *i*, where *F* is the number of features. In other words, each feature can pick exactly *m* tokens to process. We call this approach *Uniform Feature Choice*.

Uniform prevalence is a natural way to organize the features so that a feature firing provides maximal information about the token, under the assumption that all features provide approximately equal information. However, we have seen that in existing open-source SAEs, all features are not equivalently prevalent. Instead, they are approximately Zipf-law distributed. To maintain this distribution of feature density we choose $m_i \sim \text{Zipf}(\alpha, \beta)$, where Zipf represents a truncated Zipf distribution and *i* is the rank of a given feature in terms of feature density.

$$m_i = \text{Zipf}(i) \propto \frac{1}{(i + \beta)^{\alpha}} \tag{4}$$

We call the Feature Choice approach where the m_i are Zipf-distributed, *Zipf Feature Choice*, henceforth simply *Feature Choice*.

4.4 TRAINING APPROACH

Our approach is as follows:

- Given the Zipf exponent and bias hyperparameters, α and β, [6] , we use Algorithm 1 to determine the estimated feature density for each ranked feature. We use the estimated feature densities to define the threshold for dying features for the `aux_zipf_loss`.

[6]We may obtain these hyperparameters by performing a hyperparameter sweep. Alternatively, if we have trained SAE of the same dimensions, we may run inference with this SAE over a large dataset (100 million tokens) and track the number of times each feature activates as its *feature density* and each feature's relative feature density *rank*. We can then fit the (rank, feature density) pairs to a Zipf distribution and estimate the exponent and bias parameters, α and β. For GPT-2 sized residual stream activations (n=768), we find $\alpha \approx 1.0, \beta \approx 6.8$. In our experiments, we fix the exponent to exactly 1 for simplicity and we find that we may reuse similar α and β parameters across SAE widths.

- We then train Mutual Choice SAEs with both the `aux_zipf_loss` and `aux_k_loss`.
- Finally, we optionally fine-tune these SAEs with the Feature Choice activation function, adding the constraint on the number of tokens that each feature should process. Here there are no auxiliary loss terms.

The sparsifying activation functions σ_s for each approach are all TopK activations where the TopK is taken over the feature dimension for Token Choice (i.e. TopK), the batch dimension for Feature Choice and all dimensions for Mutual Choice.

5 Experimental Setup

Inputs: We train our sparse autoencoders on the layer 6 residual stream activations of GPT-2 small (Radford et al., 2019). For larger SAE widths (over 1M features) we train our sparse autoencoders on the 24th layer residual stream activations of Pythia-2.8B-deduped (Biderman et al., 2023). We use a context length of 64 tokens for all experiments. We preprocess the activations by subtracting the mean over the d_{model} dimension and normalize all inputs to unit L_2 norm. All experiments use the FineWeb dataset (Penedo et al., 2024) unless otherwise specified. We shuffle the activations for training our SAEs (as in Nanda (2023)). Experiments with Token Choice, Mutual Choice and Feature Choice SAEs are performed without feature resampling, where Standard and SAE++ SAEs are trained with resampling [7].

Hyperparameters: We tune learning rates based on Gao et al. (2024) suggestion that the learning rate scales like $\sqrt{n}$. We use the AdamW optimizer (Kingma & Ba, 2015) and a batch size of 1,536 [8]. We train each SAE for 10,000 steps or until convergence. We use a weight decay of 1e-5 and apply gradient clipping. We analyse SAE with widths from 4x to 32x larger than the size of the d_{model} dimension. We use gradient accumulation for larger batch sizes. We do not perform extensive hyperparameter sweeps.

Evaluation: After training, we evaluate autoencoders on sparsity L_0, reconstruction (MSE) and the difference on the model's final (Cross-Entropy) loss. We report a standard normalized version of the loss recovered (%). We additionally evaluate our SAEs' interpretability using Juang et al. (2024)'s automated interpretability (AutoInterp) process. We report the percentage of dead features across models.

Baselines: We compare our SAEs against Standard (ReLU) SAEs and TopK SAEs [9].

6 Results

We find that Feature Choice SAEs are a Pareto improvement upon the Token Choice TopK SAEs, as in Figure 4. Similarly Figure 5 illustrates that both Mutual Choice and Feature Choice SAEs provide better utilisation of the SAE capacity with fewer dead features than comparable SAE methods. Notably the Feature Choice SAE method results in the fewest (often zero) dead features.

SAE Type	6k latent dim	16m dim	34m dim
Standard SAE (w/ resampling)	9.0%	>90.0%	–
SAE++ (w/ resampling)	5.1%	–	64.7%
Token Choice (TopK) SAE	0.0%	7.0%	–
Feature Choice SAE (Ours)	**0%**	**0%**	**0%**

Table 1: The Feature Choice SAE maintains zero dead features at widths of up to 34 million features. This is in contrast to Standard SAEs (Bricken et al., 2023), the SAE++ (Templeton et al., 2024) and Token Choice (TopK) SAE (Gao et al., 2024) which have an increasing percentage of dead features with the SAE width.

[7] Bricken et al. (2023) describe a resampling procedure for dead features in which they reinitialise these dead features midway through training in order to reduce the number of dead features at the end of training. Models with the `aux_k_loss` do not require resampling but for models without the `aux_zipf_loss` we include resampling for stronger baselines.

[8] For the Feature Choice approach, it's important to have sufficiently large minibatch sizes so that each feature is expected to activate every few minibatches.

[9] We do not explicitly test against Gated SAEs, but Gao et al. (2024) find that TopK SAEs perform similarly or better than Gated SAEs with $1.5\times$ less compute to convergence.

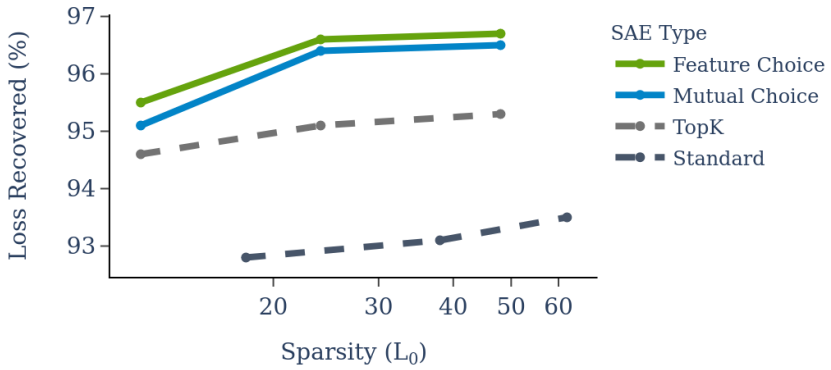

Figure 4: SAEs trained with the Mutual Choice activation function, and those finetuned with the Feature Choice activation function have up to 1.7% greater normalised reconstruction loss recovered at equivalent sparsity levels compared to TopK and standard SAEs.

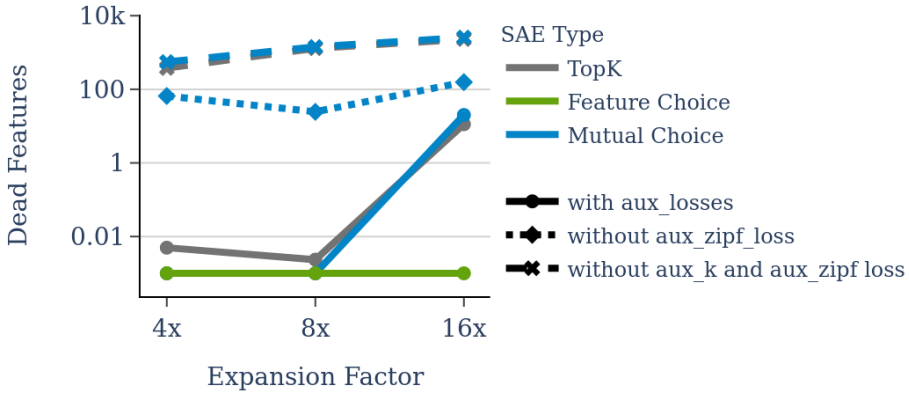

Figure 5: Both Mutual Choice SAEs and Feature Choice SAEs have fewer dead features than SAEs trained without the `aux_zipf_loss`. Even at 16× expansion factor, Feature Choice SAEs have no dead features. The `aux_zipf_loss` and `aux_k_loss` are effective at reducing the number of dead features in Mutual Choice SAEs. At large SAE widths, the `aux_zipf_loss` reduces the number of dead features by up to 13×.

Templeton et al. (2024)'s 34 million latent SAEs have a dead feature rate of 64.7% (with resampling); Gao et al. (2024)'s 16 million latent SAEs have a dead feature rate of 90% without mitigations and 7% with mitigations. We find that Feature Choice SAEs can be trained with a 0% dead feature rate for both 16m and 34m latent SAEs as in Table 1.

7 Discussion

We summarize the comparison between our approach and related approaches in Table 2:

To our knowledge, we present the first SAE training process which explicitly contains two separate phases: first *Mutual Choice training*, then *Feature Choice training*. We speculate that there may be performance benefits to phased training.

SAE Type	Performance	Dead Features	Auxiliary Losses
Standard SAE	Weak	Many	L_1 Sparsity loss
SAE++	Better	Many	Scaled L_1 sparsity loss
Jump-ReLU SAE	Better	Fewer	L_1 Sparsity loss
Token Choice (TopK) SAE	Better	Fewer	`aux_k`
BatchTopK SAE	Better	Fewer	`aux_k`
Mutual Choice SAE (Ours)	**Best**	**Fewer**	`aux_k` and `aux_zipf`
Feature Choice SAE (Ours)	**Best**	**None**	**None**

Table 2: Mutual Choice and Feature Choice SAEs (ours) achieve higher reconstruction accuracy with fewer dead features than other approaches such as Standard SAEs (Bricken et al., 2023), the SAE++ (Templeton et al., 2024), Jump-ReLU SAEs (Rajamanoharan et al., 2024b), Token Choice (also known as TopK) SAEs (Gao et al., 2024) and BatchTopK (Bussmann et al., 2024).

We believe that conceiving of the role of SAE encoders as defining matching/routing algorithms (similar to other Adaptive Computation work, for example, within the Mixture of Experts literature) could be a valuable intuition pump for further improvements to SAE architectures.

Our approaches can also be combined with the MDL-SAE (Ayonrinde et al., 2024) formulation which treats **conciseness** (**Description Length**) as the relevant quantity for evaluation and model selection rather than sparsity (L_0). The Description Length of a set of feature activations is a function of the sparsity, the distribution of activation patterns for each feature and the SAE width.

The computational efficiencies of the Token Choice (TopK), Mutual Choice and Feature Choice approaches are theoretically equivalent in terms of Floating Point Operations (FLOps) and in our experiments we find empirically that the Mutual Choice and Feature Choice activation functions do not add significant computational requirements (<1% difference in wall clock time) despite converging more quickly.

7.1 Zipf Distributed Features

We find that constraining the number of tokens per feature with the Zipf distribution outperforms using the uniform distribution by >10% model loss recovered. The large drop in performance using the uniform distribution compared to the Zipf distribution gives additional evidence that naturally occurring features are not uniformly distributed.

We hypothesise that the reason that features appear to be Zipf distributed may be strongly analogous to the reason that the frequency of words in natural languages like English are also Zipf distributed. Words are the semantic units of sentences. Since features are the semantic units of computation within language models, a similar mechanism could explain the empirical tendency for the densities of SAE features to tend towards the Zipf distribution.

In the Computational Linguistics literature, it is well known that the distribution of words in natural languages approximately follows a Zipf distribution (Zipf, 1949). For example, in written English text, the empirical distribution over words (treated as a categorical variable) can be modelled as Zipf(α, β) = Zipf(1, 2.7).

We speculate that the Preferential Attachment Theory explanation (Zhu et al., 2018; Chen, 2012) for the tendency of words to be Zipf-distributed, which states that frequently used words (features) tend to be used more often, may be analogously applicable here. At initialisation, there is some variance in feature prevalence. The tokens which are initially most highly activated early in training receive the most gradient signal and are most refined, leading to a virtuous cycle where they are more effective and useful for a larger part of the feature density spectrum. Figure 6 illustrates this dynamic over time. We would be excited about further work detailing an explicit mechanism for why features in Neural Networks tend towards being Zipf distributed.

7.2 Adaptive Computation for Varying Token Difficulty

One benefit of the Mutual Choice and Feature Choice approaches is that they allow for *Adaptive Computation* - difficult to reconstruct tokens (which may represent complex or rare concepts) can be reconstructed using more features, whereas other tokens which are more straightforward can use fewer features. Where Token Choice (TopK) SAEs suggest that all tokens are equal; MC and FC SAEs suggest that, in fact, some tokens are more equal than others Singer (2017).

As an extreme example, we might expect that the activations resulting from the $<$ BOS $>$ token are relatively easy to reconstruct, considering they are both very common and have exactly the same value every time. We might expect that an effective SAE could learn to productively reallocate the sparsity budget that would have been spent on the $<$ BOS $>$ token to more difficult tokens, thus increasing the SAE's effective capacity. We provide an example of the features per token distribution in Appendix D.

7.3 Dead Features

Our methods, especially Feature Choice SAEs, have many fewer dead features than other approaches (typically zero) without complex and compute-intensive resampling procedures (Bricken et al., 2023). This is especially important for large SAEs where the problem of dead features is typically more significant. We hypothesize that this might be an additional reason for the improved performance: all features receive some gradient signal at every step.

We note that this simplified approach eliminates the need for complex resampling procedures. This approach suggests a simplified procedure for SAE training and model selection than previous SAEs with fixed sparsity budget.

7.4 Limitations

Appeals to the Monotonic Importance Heuristic: In Section 3.1 we defined the Monotonic Importance Heuristic (MIH) - the assumption that the importance of a feature is monotonically increasing in feature activation magnitude. We use this assumption when we choose the feature activations with the largest magnitudes with our TopK-style activation functions. TopK SAEs also implicitly assume the MIH (Monotonic Importance Heuristic). We can think of Jump-ReLU SAEs and Gated SAEs as relaxing this assumption slightly to a weak MIH. For Jump-ReLU and Gated SAEs, the importance of activations is still related to their magnitude and they still filter out any low magnitude feature activations; however, the filtering threshold varies for each feature. So Jump-ReLU SAEs do not have to make magnitude comparisons across features which may have different natural scales. It may be, however, that there are low magnitude activations which, within a certain context, are nonetheless critically important in capturing information which is useful for reconstruction and/or downstream model performance. These important but low magnitude activations are difficult to capture with our current SAE approaches [10].

Though the weight-sharing form of Gated SAEs (Rajamanoharan et al., 2024a) implicitly encodes the weak MIH prior, the non-sharing form does not. Weight-sharing Gated SAEs, however, tend to perform better. The improved performance of approaches which encode the MIH prior could be considered as evidence for the truth of the claim. Alternatively, we might note that the Monotonic Importance Heuristic acts as an inductive bias for our models. Good inductive biases often allow models to perform better at first; however, with increased scale we may not need such inductive biases and may prefer allowing the model to learn more of the solution independently (Xiao, 2024), (Sutton, 2019).

Generalisation Across Modalities: We tested our SAEs within the domain of language. We currently don't know to what extent our results generalise across modalities, especially to inherently continuous modalities like audio and images. We would be excited about future work applying similar techniques to Interpretability problems in a wider range of modalities.

Evaluation: The Mechanistic Interpretability field doesn't currently have widely agreed upon metrics for evaluating Sparse Autoencoders. Disentanglement benchmarks like Huang et al. (2024) have been proposed as well as evaluation on tasks where the ground truth is known (Karvonen et al., 2024). Developing a more comprehensive suite of benchmarks for SAEs would help us to have higher confidence in our comparisons between SAE variants.

8 Conclusion

We introduce the Feature Choice and Mutual Choice SAEs as a simple drop-in change to the sparsifying activation function in Sparse Autoencoders. We also provide a new auxiliary loss, `aux_zipf_loss`, which prevents dying features and hence allows the SAE to more fully utilize all of its features without wasting capacity.

[10] Standard ReLU SAEs do allow low magnitude feature activations but at the expense of failing to filter out noisy low magnitude activations, which can be seen as false positives (Rajamanoharan et al., 2024b).

REFERENCES

Kola Ayonrinde, Michael T. Pearce, and Lee Sharkey. Interpretability as compression: Reconsidering sae explanations of neural activations with mdl-saes, 2024. URL https://arxiv.org/abs/2410.11179.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread*, 2023.

Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK: A Simple Improvement for TopK-SAEs. July 2024. URL https://www.alignmentforum.org/posts/Nkx6yWZNbAsfvic98/batchtopk-a-simple-improvement-for-topk-saes.

Yanguang Chen. The mathematical relationship between zipf's law and the hierarchical scaling law. *Physica A: Statistical Mechanics and its Applications*, 391(11):3285–3299, June 2012. ISSN 0378-4371. doi:10.1016/j.physa.2011.12.031. URL http://dx.doi.org/10.1016/j.physa.2011.12.031.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, et al. On the representation collapse of sparse mixture of experts. *Advances in Neural Information Processing Systems*, 35:34600–34613, 2022.

Truong Giang Do, Le Khiem, Quang Pham, TrungTin Nguyen, Thanh-Nam Doan, Binh Nguyen, Chenghao Liu, Savitha Ramasamy, Xiaoli Li, and Steven Hoi. HyperRouter: Towards efficient training and inference of sparse mixture of experts. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5754–5765, Singapore, December 2023. Association for Computational Linguistics. doi:10.18653/v1/2023.emnlp-main.351. URL https://aclanthology.org/2023.emnlp-main.351.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha,

Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

N. Benjamin Erichson, Zhewei Yao, and Michael W. Mahoney. Jumprelu: A retrofit defense strategy for adversarial attacks. In *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pp. 428–435, 2020.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

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. 2024. URL https://cdn.openai.com/papers/sparse-autoencoders.pdf.

Alex Graves. Adaptive computation time for recurrent neural networks, 2017. URL https://arxiv.org/abs/1603.08983.

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. Ravel: Evaluating interpretability methods on disentangling language model representations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8669–8687. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.acl-long.470.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee 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.

Caden Juang, Gonçalo Paulo, Jacob Drori, and Nora Belrose. Open Source Automated Interpretability for Sparse Autoencoder Features, July 2024. URL https://blog.eleuther.ai/autointerp/.

Adam Karvonen, Benjamin Wright, Can Rager, Rico Angell, Jannik Brinkmann, Logan Smith, Claudio Mayrink Verdun, David Bau, and Samuel Marks. Measuring progress in dictionary learning for language model interpretability with board game models, 2024. URL https://arxiv.org/abs/2408.00113.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1412.6980.

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

Honglak Lee, Chaitanya Ekanadham, and Andrew Ng. Sparse deep belief net model for visual area V2. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://papers.nips.cc/paper_files/paper/2007/hash/4daa3db355ef2b0e64b472968cb70f0d-Abstract.html.

Tianlin Liu, Mathieu Blondel, Carlos Riquelme Ruiz, and Joan Puigcerver. Routers in vision mixture of experts: An empirical study. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=aHk3vctnf1.

Julien Mairal, Francis Bach, and Jean Ponce. Sparse modeling for image and vision processing. *Foundations and Trends® in Computer Graphics and Vision*, 2014.

Alireza Makhzani and Brendan Frey. k-Sparse Autoencoders. In *International Conference on Learning Representations (ICLR)*, 2014. URL https://openreview.net/forum?id=QDm4QXNOsuQVE.

Neel Nanda. Open Source Replication & Commentary on Anthropic's Dictionary Learning Paper. October 2023. URL https://www.lesswrong.com/posts/fKuugaxt2XLTkASkk/open-source-replication-and-commentary-on-anthropic-s.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan

Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 Technical Report, March 2024. URL http://arxiv.org/abs/2303.08774. arXiv:2303.08774 [cs].

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving Dictionary Learning with Gated Sparse Autoencoders, April 2024a. URL http://arxiv.org/abs/2404.16014. arXiv:2404.16014 [cs].

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders, 2024b. URL https://arxiv.org/abs/2407.14435.

Lee Sharkey, Dan Braun, and beren. [Interim research report] Taking features out of superposition with sparse autoencoders. December 2022. URL https://www.alignmentforum.org/posts/z6QQJbtpkEAX3Aojj/interim-research-report-taking-features-out-of-superposition.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017. URL https://openreview.net/pdf?id=B1ckMDqlg.

Peter Singer. All animals are equal. In *Animal rights*, pp. 3–16. Routledge, 2017.

Richard Sutton. The Bitter Lesson, 2019. URL http://www.incompleteideas.net/IncIdeas/BitterLesson.html.

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.

Lechao Xiao. Rethinking conventional wisdom in machine learning: From generalization to scaling, 2024. URL https://arxiv.org/abs/2409.15156.

Fuzhao Xue, Valerii Likhosherstov, Anurag Arnab, Neil Houlsby, Mostafa Dehghani, and Yang You. Adaptive computation with elastic input sequence. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38971–38988. PMLR, 2023. URL https://proceedings.mlr.press/v202/xue23e.html.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35: 7103–7114, 2022.

Yueying Zhu, Benwei Zhang, Qiuping A. Wang, Wei Li, and Xu Cai. The principle of least effort and zipf distribution. *Journal of Physics: Conference Series*, 1113(1):012007, nov 2018. doi:10.1088/1742-6596/1113/1/012007. URL https://dx.doi.org/10.1088/1742-6596/1113/1/012007.

George K. Zipf. *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, 1949.

A Feature Density Moves Towards A Zipf Distribution Throughout Training

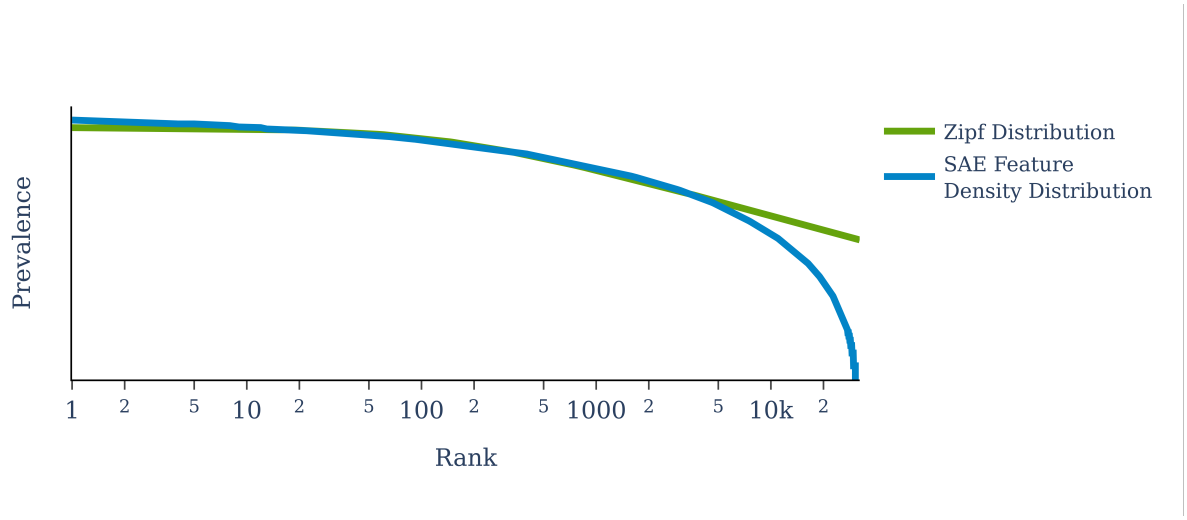

Figure 6: For an untrained SAE at initialiation, the Feature Density Distribution is less well fitted to the Zipf distribution than after training. Whilst in Figure 3, we see the Feature Distribution diverge from the Zipf distribution at the 25,000th ranked feature, here we see divergence from the 5,000th ranked feature.

Compared to after training (see Figure 3), the Feature Density Distribution diverges from the Zipf distribution at a much earlier token rank for the features at initialisation. We empirically see that over the course of training, the feature distribution approaches the Zipf distribution more closely. This convergence over training further suggests that the Zipf distribution might be a natural distribution for features. We see this pattern over multiple datasets as shown in Table 3

Dataset	R^2
Wikipedia	0.982
FineWeb	0.983
Arxiv Abstracts	0.986
Biology Arxiv Abstracts	0.984
ML ArXiv Abstracts	0.986

Table 3: Across a range of datasets, the R^2 correlation between the Feature Density Distribution and Zipf Distribution is consistently >0.982 suggesting that the findings that features are Zipf distributed is a general phenomena.

B Inference with Feature Choice and Mutual Choice SAEs

To run inference on our SAE variants, we may perform batch inference with the method exactly as in the training setup. However to do single token (or single sequence) inference (i.e. in the low batch size regime) it may be beneficial to instead impute a threshold value and swap out the activation function to use this value instead with a JumpReLU style approach (Erichson et al., 2020) (Rajamanoharan et al., 2024b).

C Neural Feature Matrix Loss

One problem with SAEs is undesirable feature splitting. Feature splitting occurs when an SAE finds a sparse combination of existing directions that allows for a smaller L_0 (Ayonrinde et al., 2024). For example, Bricken et al. (2023) note that a model may learn dozens of features which all represent the letter "P" in different contexts in order to maintain low sparsity.

In order to reduce feature splitting we propose two additional auxiliary losses: `nfm_loss` and `nfm_inf_loss`.

The Neural Feature Matrix (NFM) is defined as $\texttt{nfm}(\mathbf{W}) = \hat{\mathbf{W}}\hat{\mathbf{W}}^T$ for a weight matrix $\mathbf{W}$. The NFM is a symmetric square matrix which describes the correlation of different rows in the matrix $\mathbf{W}$.

We define $\texttt{nfm\_loss} = |\texttt{nfm}(\mathbf{W}_{Dec})|_F$ and $\texttt{nfm\_inf\_loss} = \frac{1}{F}\sum_i max(\texttt{nfm}(\mathbf{W}_{Dec})_i)$.

For our largest runs we apply both of these auxiliary losses with small weight. Empirically we find this seems to reduce undesirable feature splitting and avoid a failure mode we call "Dictionary Collapse" when many features of the decoder dictionary start to align with each other. The Dictionary Collapse phenomena appears to be closely analogous to the Representation Collapse problem in Sparse Mixture of Experts (SMoE) models as detailed in Chi et al. (2022) and Do et al. (2023).

D Adaptive Computation Allows For A Variable Number of Features Per Token

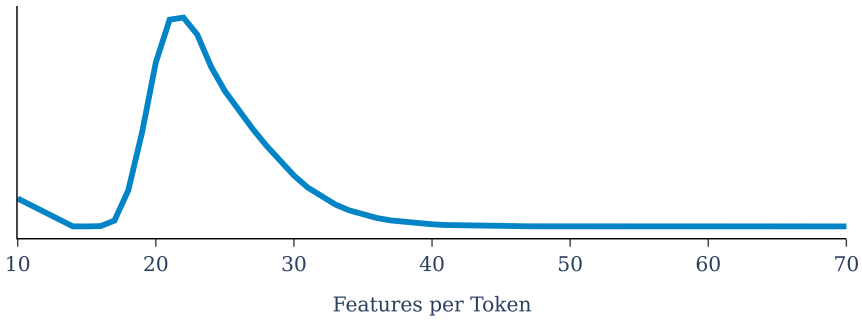

Figure 7: For Mutual Choice SAEs, the distribution of active features per token is bimodal with a long tail at the upper side. The SAE is able to allocate more features and more computation to more difficult to reconstruct features and perform Adaptive Computation.

We note that there is a bimodal distribution of features per token (see Figure 7). The SAE allocates close to the main mode of features to each token, attempting to allocate more features to harder tokens and less features to easier tokens. The $<$ BOS $>$ token is responsible for the lower peak - this token is much easier to reconstruct and so it is prudent for the network to allocate fewer features that token. In the ideal SAE, the $<$ BOS $>$ token could be reconstructed with a single feature, rather than the 10 features required here. Future work might look into optimisations that would allow the $<$ BOS $>$ token to be single-feature reconstructed as a test case of allowing even greater variance in features per token.

E Progressive Codes

Gao et al. (2024) describe learning a progressive code using their Multi TopK loss. In their setting, the Multi TopK loss (a weighted sum of TopK losses for different values of k) is required because the SAE generally "overfits" to the value of k which harms a progressive code. In our case, the SAE is robust to having variable k values even for the same token depending on the context of the batch. Empirically, we obtain progressive codes for a greater range of values of k than in the TopK case.

F Monotonic Importance Heuristic

In Section 3 we appeal to the Monotonic Importance Heuristic (MIH), as in TopK SAEs, in order to simplify our allocation problem. Empirically we find that this works well though we discuss the case for not using the MIH in Section 7.4.

We can formally write the Monotonic Importance Heuristic as the hypothesis that given some token-feature affinities $\mathbf{z}$' and corresponding feature activations $\mathbf{z_1}$ and $\mathbf{z_2}$, if $\|\mathbf{z_1}\|_1 > \|\mathbf{z_2}\|_1$, $\mathbf{z_1}$ is likely to result in lower reconstruction error under the decoder map.

One theoretical (though informal) motivation for the MIH is as follows. Since the decoder dictionary is fixed to unit norm, the norm of any feature's contribution to the output is exactly the magnitude of the feature activation to which it corresponds. Hence if there's limited cancellation between features (which is likely in an N dimensional space where the per-token sparsity is much less than N) then we might expect the component of $\hat{\mathbf{x}}$ in any feature direction to be very close to the feature activation for that feature. In particular, consider a feature with a small magnitude of ε. This feature can only possibly influence the reconstruction loss by at most $\varepsilon|\mathbf{e}|$ where $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$. Features corresponding to larger activations can plausibly influence the reconstruction loss by a greater amount.

G Determining The Feature Distribution

Given the Zipf exponent and bias hyperparameters, α and β, we use Algorithm 1 to determine the estimated feature density for each ranked feature. We use the estimated feature densities to define the threshold for dying features for the `aux_zipf_loss`.

Algorithm 1 Calculate Zipf Feature Distribution

Require: k, F, B, β, α, m_{max}
Ensure: m: array of size F
num_interactions $\leftarrow B \times k$
zipf_sum $\leftarrow \sum_{i=1}^{F} \frac{1}{(i+\beta)^{\alpha}}$
$N_{\text{approx}} \leftarrow \frac{\text{num\_interactions}}{\text{zipf\_sum}}$
for $i = 1$ **to** F **do**
    $m_i \leftarrow \min\left(\left\lfloor \frac{N_{\text{approx}}}{(i+\beta)^{\alpha}} \right\rfloor, m_{\text{max}}\right)$
end for
return m

H Optimality of TopK Activation Under the Monotonic Importance Heuristic

Proposition 1. *Let* $\mathbf{z} = \{z_1, z_2, ..., z_F\} \in \mathbb{R}^F$ *and consider* $f_z(\boldsymbol{a}) = \boldsymbol{a}^T \cdot \boldsymbol{z}$*, where* $\boldsymbol{a} = \{a_1, a_2, ..., a_F\} \in \{0,1\}^F$ *is a k-sparse boolean vector and hence* $\boldsymbol{a}$ *satisfies* $\sum_{i=1}^{F} a_i \leq k$*.*

Then for any optimal solution $\boldsymbol{a}^*$*, if* $a_i^* = 1$ *then there exist at most* $k-1$ *indices* j *such that* $z_j > z_i > 0$*.*

Proof. We proceed by contradiction. Suppose there exists some optimal solution **a*** and an index i with $a_i^* = 1$ such that there are $m > k-1$ indices z_j with $z_j > z_i > 0$. Since **a*** is k-sparse, there must exist one such index J such that $z_J > z_i > 0$ and $a_J^* = 0$.

Consider **a'** constructed as follows:

$$a'_j = \begin{cases} a_j^* & \text{if } j \neq i \text{ and } j \neq J \\ 0 & \text{if } j = i \\ 1 & \text{if } j = J \end{cases} \tag{5}$$

First note that **a'** remains a k-sparse boolean vector as we have only swapped two elements. We see this as **a'** $\in \{0,1\}^F$ and both **a'** and **a*** have the same number of non-zero elements by construction.

We now note that $f_z(\mathbf{a'}) - f_z(\mathbf{a^*}) = z_J - z_i > 0$, where the inequality follows from our choice of J. Hence we have some feasible **a'** with $f_z(\mathbf{a'}) > f_z(\mathbf{a^*})$.

This contradicts our assumption that $f_z(\mathbf{a^*})$ was optimal. □

In other words, suppose we assume the Monotonic Importance Heuristic (MIH) (Section 3.1, Appendix F) on some set of real numbers, **z**. If more than k elements of **z** are positive, then it follows that the optimal activation function to maximise the sum of the activated affinities (and hence by the MIH, to maximise the reconstruction accuracy) under the k-sparsity constraint, is the TopK activation function. For the case where fewer than k elements are positive, optimality is achieved by activating only the positive elements, as including any negative values would decrease the sum. Therefore, the composite activation function TopK $\circ$ ReLU provides the optimal solution in full generality.