HOW OPTIMALITY STRUCTURES SPARSE DICTIONARIES THEORY FOR INTERPRETING SAE REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse Autoencoders (SAEs) have found success parsing neural network representations into interpretable concepts, providing a basis for understanding and control. However, what exactly SAEs extract and, hence, the scientific conclusions we can draw from them are not obvious. In short, if your SAE behaves strangely, does that reflect interesting neural network behaviour or an SAE-imposed distortion? Towards answering this, we use dictionary learning identifiability results to derive constraints that optimal dictionary learning features must satisfy. For example, an optimal feature will never turn on only while another is active. We use these conditions to explain various SAE oddities - hierarchical splitting & absorption, which features can be left in the residuals, dense antipodal features, and infinite feature splitting - simply as properties imposed by the dictionary learning objective. Finally, these constraints are diagnostic: real SAEs pass when measured on the dataset on which they were trained, but increasingly fail as the test dataset becomes more 'distant'. In sum, we hope to provide theoretical tools to explain puzzling SAE patterns, allowing more principled inferences about internal model behaviour.

1 INTRODUCTION

Dictionary learning methods model data using a sparse linear combination of learnt components. They offer practical algorithms that operationalise the intuition that the world is composed of parts (a chair, the sun, twilight, ...) each contributing to a small portion of experience, and learns these parts from observations. They have found widespread use, both practically (Mairal et al., 2008; Wright et al., 2008; Elad, 2010; Dumitrescu & Irofti, 2018), and as a normative model of brains' sensory processing (Olshausen & Field, 1996; 2004; Sadagopan & Wang, 2009; Willmore & King, 2009).

More recently, mechanistic interpretability, which tries to understand neural networks as a neuroscientist does the brain, has driven a surge of interest in dictionary learning as a tool to understand neural networks. This was motivated by the finding that neural networks often encode interpretable concepts as linear directions (Mikolov et al., 2013; Arora et al., 2016). Further, these concepts are often sparse, and there appear to be more of them than the representation has neurons, forcing the network to encode them in 'superposition' (Arora et al., 2018; Elhage et al., 2022). While superposition can hinder single neuron interpretation, conceptualising representations as sparse linear combinations of concepts - the 'linear representation hypothesis' - naturally motivates using dictionary learning to recover the concepts. Indeed, flavours of Sparse Autoencoders (SAEs), a dictionary learning variant, have proven effective at extracting meaningful concepts from LLM representations (Bricken et al., 2023; Cunningham et al., 2023; Templeton et al., 2024), and these concepts have proved capable of controlling network output (Templeton et al., 2024; Durmus et al., 2024; Arad et al., 2025).

What can the success of SAEs teach us about how neural networks represent and manipulate information? At first blush, it seems a ringing endorsement for the 'linear representation hypothesis': one can find linearly encoded concepts that control the network; this surely reflects models' internal use of these same vectors? However, a growing body of empirical work has produced puzzling observations. For example, training SAEs of increasing size (i.e. with more and more concepts) leads to 'feature splitting': a concept identified by a smaller SAE, such as 'base64', splits into multiple in larger SAE, like 'digits in base64', 'letters in base64' and 'ASCII characters in base64' (Bricken et al., 2023). This is perhaps surprising; a hierarchical decomposition in which 'base64' and 'ASCII in base64' coexist seems natural. Is feature splitting evidence that neural networks don't encode hierarchies?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Thankfully, the answer is likely 'no'. As we shall discuss, dictionary learning cannot help splitting hierarchies. This illustrates our motivation: dictionary learning assumes representations can be well-modelled as sparse linear combinations of parts. When we apply dictionary learning techniques, like SAEs, to neural networks we are not just seeing symptoms of neural network behaviour - but also the assumptions baked into our tools. Further, the way those assumptions manifest is often surprising. As such, a better understanding of our tools will allow us to more accurately link our observations to network behaviour, and provide targets for developing tools that more precisely reflect our goals.

Here, we take steps towards such an understanding. Our main result is conditions that a representation must obey to be a dictionary learning optima, analogous to earlier work in slightly different settings (Gribonval & Schnass, 2010). These conditions are interpretable, describing how optimal features and residuals are allowed to relate to one another. We use these constraints to explain SAE behaviours - hierarchical splitting & absorption, the structure of residuals, and dense antipodal features. Additionally, we highlight a wide-convex reformulation of the dictionary learning problem which shows that widening SAEs will refine their categorisations until each ray of data is assigned a concept (Bach et al., 2008). Finally, we show our conditions are diagnostic: real SAEs pass when measured on the dataset on their training set, but progressively fail for more distant datasets. In sum, we provide tools for understanding the behaviour of dictionary learning, and use it to relate otherwise puzzling SAE properties to the interaction between data and modelling assumptions. In so doing, we hope to motivate alternative interpretability approaches that reflect more realistic representational structures.

2 RELATED WORK: 3 RELATED THREADS

Dictionary Learning Identifiability There has been extensive work on the identifiability of dictionary learning, drawing on earlier work in compressed sensing (sparse coding with a fixed dictionary) (Candes & Tao, 2006; Donoho, 2006; Donoho & Elad, 2003), and matrix factorisation (Tatli & Erdogan, 2021; Gillis, 2020). Much of it assumes a particular data-generating process, then describes conditions under which the true factors are recovered (Geng & Wright, 2014; Schnass, 2015; Hillar & Sommer, 2015; Wu & Yu, 2018; Gribonval et al., 2015; Cohen & Gillis, 2019; Wang et al., 2020). Here, motivated by the heterogeneity of neural network representations, we focus on approaches that guarantee optimality without data-generating assumptions. There exist global identifiability results of this generality (Hu & Huang, 2023), but they require novel formulations of the problem. For more standard dictionary learning problems Gribonval & Schnass (2010) derive local optimality conditions without data assumptions, and we use these techniques in our work. Further, we use the results of Bach et al. (2008) to reformulate dictionary learning and study the large width limit. Relative to this literature our novelty lies in interpreting a set of conditions similar to those of Gribonval & Schnass (2010) and making them do work to explain the behaviour of SAEs.

SAE Theory We highlight three theoretical approaches to understanding SAE representations. First, two recent works study the conditions under which SAEs recover sparse features: Cui et al. (2026) show recovery under extreme sparsity of features whose embedding dot-products are nonpositive, while Tang et al. (2025) study an approximate loss valid under sufficient sparsity and show that it is minimised only by true feature recovery. By making specific dataset assumptions, these are analogues of older dictionary learning work. We instead study constraints on generic dictionary learning optima allowing us to study the representation of data that doesn't respect sparse data-generating assumptions. Second, Hindupur et al. (2025) study how architectural constraints (e.g. choosing TopK or JumpReLU SAE (Gao et al., 2024; Rajamanoharan et al., 2024)) determine the concepts SAEs can extract. By studying vanilla dictionary learning our work ignores such architectural effects. Instead we focus on the distributional assumptions in SAEs, a relatively under-explored aspect. Third, follow-up work on feature splitting and absorption linked these phenomena to the interplay of sparsity and hierarchy (Till, 2024; Chanin et al., 2024). We build on these works, making the same link more robustly.

Neural Coding There's decades of neuroscientific work relating representations to solutions of optimisation problems (Laughlin, 1981; Atick & Redlich, 1990; Simoncelli & Olshausen, 2001), including, famously, dictionary learning (Olshausen & Field, 1996). We draw particular technical inspiration from Sengupta et al. (2018); Dorrell et al. (2026) who show the convexity of nonnegative neural coding problems by lifting to the space of representational similarity matrices. This is a particular example of the semi-definite relaxation, common in polynomial optimisation (Shor, 1987; Lasserre, 2009; Anstreicher, 2012; Pena et al., 2015) or matrix factorisation (Bach et al., 2008).

3 PROBLEM, REFORMULATION, LIMITING CONVEXITY, & OUTLINE

In this section we outline the version of dictionary learning we study and provide novel reformulations, one entirely in terms of the latent code and another that is convex for wide dictionaries.

Dictionary Learning & Notation The basic assumption of both dictionary learning and the linear representation hypothesis is that data, $\boldsymbol{x}^{[i]} \in \mathbb{R}^{d_x}$ - in this case, a neural network representation - can be well modelled as a linear combination of a set of sparse components: $\boldsymbol{x}^{[i]} \approx \boldsymbol{W}\boldsymbol{z}^{[i]} + \boldsymbol{b}$. Here, $\boldsymbol{z}^{[i]} \in \mathbb{R}^{d_z}$ is the (hopefully) sparse SAE representation of datapoint i, $\boldsymbol{W}$ is the weight matrix/dictionary with columns $\boldsymbol{w}_d$ that approximately maps $\boldsymbol{z}^{[i]}$ to $\boldsymbol{x}^{[i]}$, $\boldsymbol{b}$ is a bias vector, d_x and d_z are the size of the neural network representation and the SAE respectively (generally $d_z \gg d_x$), and i runs from 1 to N, the number of datapoints. Further, we will denote with $\boldsymbol{z}_d \in \mathbb{R}^N$ the vector of responses of SAE neuron d. To learn this decomposition we optimise the following loss:

$$\min_{\boldsymbol{W},\{\boldsymbol{z}^{[i]}\}_{i=1}^N,\boldsymbol{b}} \mathbb{E}_i\left[||\boldsymbol{W}\boldsymbol{z}^{[i]} + \boldsymbol{b} - \boldsymbol{x}^{[i]}||_2^2 + 2\lambda||\boldsymbol{z}^{[i]}||_1\right], \qquad \text{subject to} \quad \boldsymbol{z}^{[i]} \geq 0, \quad ||\boldsymbol{w}_d||_2 = 1 \tag{1}$$

The objective has two terms, the reconstruction error that ensures accurate fitting of the data, and an L1 penalty that encourages sparseness (we define λ with a 2 for later convenience). Without constraints, a trivial approach to decreasing the loss involves simultaneously scaling up $\boldsymbol{W}$ and down $\boldsymbol{z}^{[i]}$. Following Bricken et al. (2023), we control this by constraining each dictionary column, $\boldsymbol{w}_d$, to be unit norm. Finally, $\boldsymbol{z}^{[i]} \geq 0$ is a common but non-essential choice that reflects the idea that components should contribute in one direction. In the following, we will describe constraints on feature directions, $\boldsymbol{w}_d$, and distributions, $\boldsymbol{z}_d$, implied only by optimising eq. (1).

Encoder Architecture While linearity from $\boldsymbol{z}^{[i]}$ to $\boldsymbol{x}^{[i]}$ reflects the key representational hypothesis, the encoder from $\boldsymbol{x}^{[i]}$ to $\boldsymbol{z}^{[i]}$ is less specified. Standard SAEs learn a separate affine+ReLU transformation: $\boldsymbol{z}^{[i]} = \text{ReLU}(\boldsymbol{W}_{\text{enc}}\boldsymbol{x}^{[i]} + \boldsymbol{b}_{\text{enc}})$ (Bricken et al., 2023; Cunningham et al., 2023; Templeton et al., 2024), which additionally constrains the learnt concepts. Motivated partly by this gap between ideal dictionary learning representations, i.e. solutions to eq. (1), and those learnt by a standard SAE, various alternate encoders (Hindupur et al., 2025), losses (Fel et al., 2025), or nonlinearities (Gao et al., 2024; Bussmann et al., 2024; Rajamanoharan et al., 2024) have been considered, including sparse coding algorithms which can close this gap (Costa et al., 2025; O'Neill et al., 2024).

Despite this architectural diversity, we study solutions to the vanilla dictionary learning problem, eq. (1), equivalent to encoders with no optimisation gap (O'Neill et al., 2024). We view this as an idealised theoretical target demonstrating how the bare bones of dictionary learning suffice to heavily structure optimal representations. That said, some of our results, e.g. the following reformulations, hold if we add the encoder constraint, $\boldsymbol{z}^{[i]} = \text{ReLU}(\boldsymbol{W}_{\text{enc}}\boldsymbol{x}^{[i]} + \boldsymbol{b}_{\text{enc}})$, and there are certain settings in which standard SAEs exactly solve eq. (1) (see app. A.1).

Loss Reformulation In app. A.2, we use a scaling symmetry argument from Bach et al. (2008) to develop two equivalent formulations of the original problem, eq. (1). First, we show that, for the following problem studied by Templeton et al. (2024) in which the norm of the dictionary elements are regularised rather than constrained, the solutions are equivalent up to scaling:

$$\min_{\boldsymbol{W},\{\boldsymbol{z}^{[i]}\}_{i=1}^N,\boldsymbol{b}} \mathbb{E}_i\left[||\boldsymbol{W}\boldsymbol{z}^{[i]} + \boldsymbol{b} - \boldsymbol{x}^{[i]}||_2^2\right] + 2\lambda\sum_{d=1}^{d_z}||\boldsymbol{w}_d||_2\mathbb{E}_i[z_d^{[i]}], \qquad \text{subject to} \quad \boldsymbol{z}^{[i]} \geq 0 \tag{2}$$

Hence, while changing from eq. (1) to eq. (2) aids optimisation (Templeton et al., 2024), it does not change the optima, a point likely well-understood by the original authors, though not stated.

A second reformulation related by scaling uses the same trick to separate activity and weight norms:

$$\min_{\boldsymbol{W},\{\boldsymbol{z}^{[i]}\}_{i=1}^N,\boldsymbol{b}} \mathbb{E}_i\left[||\boldsymbol{W}\boldsymbol{z}^{[i]} + \boldsymbol{b} - \boldsymbol{x}^{[i]}||_2^2\right] + \frac{\lambda}{N}\sum_{d=1}^{d_z}(||\boldsymbol{w}_d||_2^2 + ||\boldsymbol{z}_d||_1^2), \qquad \text{subject to} \quad \boldsymbol{z}^{[i]} \geq 0 \tag{3}$$

This loss is an unconstrained quadratic in the dictionary, $\boldsymbol{W}$, so we can optimise it out, app. A.3. Denoting with $\boldsymbol{Z} \in \mathbb{R}^{d_z \times N}$ and $\boldsymbol{X} \in \mathbb{R}^{d_x \times N}$ the matrices formed by stacking SAE and network representations respectively, and $\bar{\boldsymbol{Z}}$ and $\bar{\boldsymbol{X}}$ their row-demeaned versions, the optimal dictionary is $\boldsymbol{W}^* = \bar{\boldsymbol{X}}(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{Z}}^T$, giving our third reformulation solely in terms of the latent code:

$$\min_{\boldsymbol{Z}} \text{Tr}[(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}] + \mathbf{1}^T\boldsymbol{Z}^T\boldsymbol{Z}\mathbf{1}, \quad \text{subject to} \quad \boldsymbol{Z} \geq 0 \tag{4}$$

Convex Reformulation In app. A.4, we note that the problem, eq. (4), can be written entirely in terms of the representational similarity, $\boldsymbol{Q} = \boldsymbol{Z}^T\boldsymbol{Z} \in \mathbb{R}^{N\times N}$, and its demeaned version $\bar{\boldsymbol{Q}}$:

$$\min_{\boldsymbol{Q}} \mathrm{Tr}[(\lambda\mathbb{1} + \bar{\boldsymbol{Q}})^{-1}\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}] + \mathbf{1}^T\boldsymbol{Q}\mathbf{1}, \quad \text{subject to} \quad \boldsymbol{Q} \in \mathcal{CP}^N, \quad \text{cp-rank}(\boldsymbol{Q}) \leq d_z \tag{5}$$

$\boldsymbol{Z} \geq 0$ has become the constraint that $\boldsymbol{Q} = \boldsymbol{A}^T\boldsymbol{A}$ for some $\boldsymbol{A} \geq 0$. Matrices with this property are called completely positive, denoted by $\boldsymbol{Q} \in \mathcal{CP}^N$ (Berman & Shaked-Monderer, 2003) and form a convex set. Further, the objective is a convex function of $\boldsymbol{Q}$ (Dorrell et al., 2026). The remaining non-convexity is the restriction cp-rank$(\boldsymbol{Q}) \leq d_z$ on the completely positive rank of $\boldsymbol{Q}$, the minimal number of rows of $\boldsymbol{A} \geq 0$ such that $\boldsymbol{Q} = \boldsymbol{A}^T\boldsymbol{A}$. The cp-rank of an $N \times N$ matrix is bounded by $N(N+1)/2$ (Shaked-Monderer et al., 2013), so as long as $d_z \geq N(N+1)/2$ we can drop this and the optimisation is convex, ensuring local minima are global. Hence, in the unrealistic very wide setting, the dictionary learning problem can be reformulated as convex. While interesting, determining membership of $\mathcal{CP}^N$ is NP-Hard (Dickinson & Gijben, 2014) making this more useful theoretically than practically. We will use this framing to study the wide-dictionary limit in section 7.

Vanilla SAEs Finally, by adding the encoding constraint $\boldsymbol{z}^{[i]} = \mathrm{ReLU}(\boldsymbol{W}_{\text{enc}}\boldsymbol{x}^{[i]} + \boldsymbol{b}_{\text{enc}})$, each of the above problems can be turned into its vanilla SAE analogue. Pleasingly, thanks to the homogeneity of ReLU, the preceding arguments continue to hold, including convexity over $\boldsymbol{Q}$ (Dorrell et al., 2026).

Outline Our analyses now uses these different reformulations to explore a series of phenomena:

- In section 4, we present a particular first-order optimality condition constraining the relationships between optimal features. We use this to identify separable feature sets, explain hierarchical splitting and absorption, and broadly get a handle on SAE feature preferences.
- In section 5 we present similar first-order optimality conditions relating features to residuals, constraining the relationship between what an optimal dictionary does and doesn't encode.
- In section 6 we turn to the finding of dense antipodal pairs of features, and explain this using a completely solvable toy model of rank 1 data.
- In section 7 we study the wide convex limit of dictionary learning, and find that data is split into rays, reinforcing the width-dependence of discovered features.

4 FEATURE CONSTRAINTS, AND HIERARCHICAL SPLITTING/ABSORPTION

Here, we describe how optimality forces features to inactivate separately, explaining feature splitting.

Local Optimality Conditions If optimised correctly, a dictionary learning solution will converge to a local minima. Given just local optimality, what can we say about the representation, $\boldsymbol{Z}$? It turns out, quite a lot. Being locally optimal means there is no perturbation that, while preserving nonnegativity, decreases the loss to first order. In app. B we study perturbations and, using either the subgradients of non-differentiable functions or dual feasible certificates, derive conditions on $\boldsymbol{Z}$ such that no loss-decreasing perturbation exists, as in Gribonval & Schnass (2010). We present the general perturbation in app. B.2, which recovers the classic ISTA-like algorithm (Daubechies et al., 2004) and gets the closest to a sufficient condition. Here, we consider a more specialised result that constrains the relationship between learnt features, and in section 5 we consider feature-residual relations.

Necessary Feature Relations In app. B.3, we study a perturbation within the span of the features, $\boldsymbol{Z}$. From this we derive the following necessary condition relating features to one another.

$$-\hat{\boldsymbol{W}}^T\hat{\boldsymbol{w}}_d \equiv -\cos(\boldsymbol{\theta}_d) \in \text{Convex Hull}\left(\left\{\frac{\bar{\boldsymbol{z}}^{[i]}}{|\min_j \bar{\boldsymbol{z}}^{[j]}|}\middle| z_d^{[i]} = 0\right\}\right) \tag{6}$$

where the $\min$ and division are taken elementwise and $\hat{\boldsymbol{w}}_d$ denotes the unit-scaled $\boldsymbol{w}_d$. Since Gribonval & Schnass (2010) consider a complete, basis-forming dictionary, this is an exact analogue of their result in our slightly different problem setting. The left hand side is a d_z-dimensional vector of cosine similarities between $\boldsymbol{w}_d$ and each of the other dictionary elements. The equation states that this vector of cosines must lie within the convex hull of the normalised data observed when feature d is off. This enforces a relationship between optimal features' directions and joint-distributions.

Intuition Consider a 2-element dictionary and with only one cosine similarity: $\cos(\theta_{12}) = \hat{\boldsymbol{w}}_1^T\hat{\boldsymbol{w}}_2$. The red regions in fig. 1A illustrate the relevant convex hulls for two independent sparse variables:

the ranges that each variable explores while the other is inactive. In this case the convex hulls span the entire $[-1, 1]$ region, so every choice of $\cos(\theta_{12})$ is allowed under eq. (6), reflecting the fact that sparse, independent features are stably extracted by dictionary learning. However, it also motivates representations that won't be local optima. Consider two features that never inactivate together, fig. 1B; the convex hull will lose its minimal region, highlighted in purple, destabilising features encoded along highly aligned directions. Roughly, aligned encoding directions reflect the fact these features are explaining similar data, and since they don't inactivate together they can be merged while maintaining nonnegativity without incurring too much cost. Alternatively, two features can be very highly correlated while still satisfying eq. (6), as long as they activate and inactivate together, fig. 1C. fig. 1D shows feature recovery when training a two element dictionary on these datasets for different values of $\cos(\theta_{12})$: the features are stable only when the convex hull condition is satisfied.

Acceptable Features Which kinds of features pass eq. (6)'s conditions? The convex hull construction, e.g. fig. 1C, emphasises the role of data range. Indeed, a sufficient condition for satisfying eq. (6) *for any set of encoding directions* is that while each feature is off the others explore the extremities of their range, as in fig. 1A & C. This dataset property is a variant of range-independence, which drives feature recovery in other settings (Roth et al., 2023; Dorrell et al., 2025); here too we find dictionary learning stabilises features in which turning one element off leaves the range of the others unchanged. Alternatively, categorical features ('mammal', 'fish', etc.) are stable when orthogonally embedded, and if they are all inactivate at least half the time (i.e. no more than half the tokens are animals) they are stable for all encoding geometries, app. C, fig. 1E. Finally, in app. C we show these conclusions generalise to separable data, a generalisation where categories are no longer mutually exclusive, but in which each feature activates alone in at least one datapoint (Donoho & Stodden, 2003). In conclusion, range-independent or categorical variables can easily pass.

Hierarchical Splitting & Absorption Two striking SAE observations are feature splitting, and its cousin, feature absorption. In feature splitting, a broad concept that appears in a narrow SAE, like

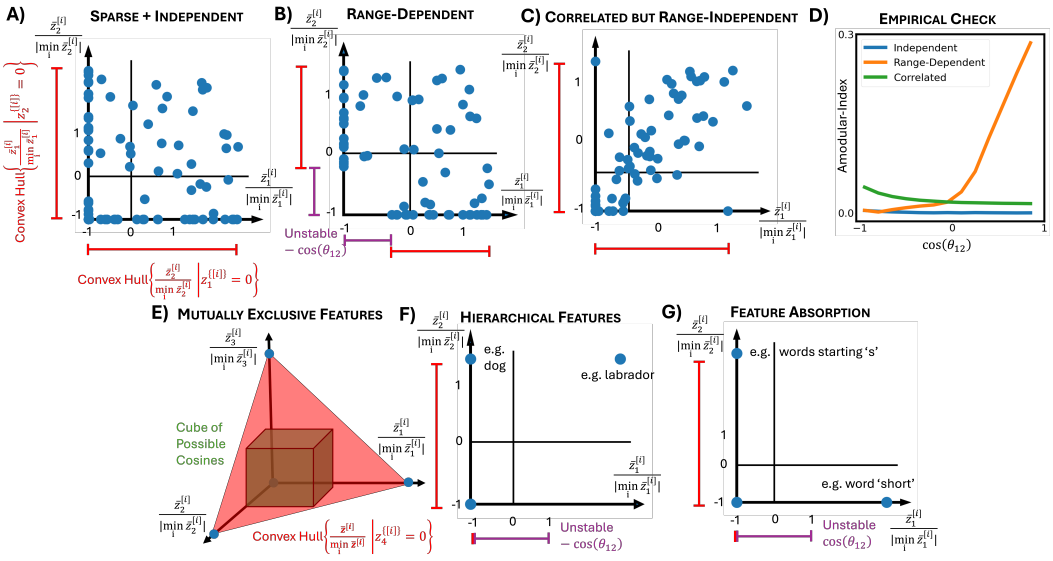


Figure 1: **A)** We plot two putatively optimal sparse independent features, normalised according to eq. (6). The relevant convex hulls (red regions) are the region explored by one feature while the other is inactive. In this case the convex hulls cover $[-1, 1]$, so these features are stable for all $\cos(\theta_{12})$. **B)** Features that never inactivate together are unstable when positively aligned (purple region), while **(C)** correlated features can be easily stabilised by points that expand the convex hull. **D)** We train a two element dictionary on the datasets in **A-C** with various alignments $\cos(\theta_{12})$. We compute an index that measures how amodular the representation is (app. C.4, 0 = each latent has its own dictionary element). We see meaningful amodularity only when predicted from the convex hulls, app. C.4. **E)** A categorical feature that is more than half sparse is stable independent of the encoding geometry. **F)** Hierarchical features are unstable under all cosine vectors except perfect alignment, whereas **(G)** feature absorption turns the hierarchy into mutually exclusive features, stabilising them.

'maths', is seen to split into more fine-grained concepts in larger SAEs, like 'geometry' or 'topology' (Bricken et al., 2023). Feature absorption, on the other hand, describes how in a single SAE with coexisting overlapping high and low level features, such as 'word starting with S' and the word 'short', the high level feature unexpectedly won't co-activate with the lower (Chanin et al., 2024). Both of these fly in the face of intuition: feature absorption creates unintuitive gaps in the activity of the higher level feature (a 'starting with S' feature that doesn't activate to the word 'short'), while feature splitting ensures SAEs miss the wood for the trees (Leask et al., 2025). Indeed, there has been SAE development motivated by exactly these frustrations (Bussmann et al., 2025; Costa et al., 2025).

Existing work has outlined the pressures that lead to this phenomenon. Till (2024) argues that, with enough neurons, an SAE should encode feature combinations rather than the features themselves, while Chanin et al. (2024) show mathematically that if the low level feature activates only when the high level feature is active, feature absorption decreases the loss. Here, we present these as natural conclusions of optimal feature-feature conditions, eq. (6), relating and generalising the findings.

To see this, consider the joint distribution of two hierarchically related features, which we operationalise as a lower level feature, such as 'labrador', that activates only when a higher level, such as 'dog', is active fig. 1F. Since all labradors are dogs, the distribution of 'labrador' while 'dog' is inactive, one of the convex hulls appearing in eq. (6), is a single point where both features are inactive. This ensures that, unless they align completely making them identical, no matter their cosine similarity, hierarchically related features are unstable and will not appear in optimal representations!

Feature splitting stabilises hierarchically related features by constructing a feature for each lower-level feature. This makes the features mutually exclusive fig. 1E, exactly those that dictionary learning naturally stabilises. Finally, feature absorption can be seen as a similar route to stability: by absorbing the instances of co-activity into the lower-level feature, the representation has again been turned into a set of mutually exclusive categories, fig. 1G. Feature absorption is simply feature splitting but using a lower-level category corresponding to 'not one of the other low-level categories'.

Real SAEs We tested whether eq. (6) is satisfied for vanilla SAEs trained on block 8 of GPT2-small's residual stream using OpenWebText (OWT) (Bloom, 2024; Bloom et al., 2024), and found no pairwise violations among 3700 sufficiently active features (14M pairs) nor in sets of up to 100 sampled features, app. D. To show that these conditions are not vacuous, we apply the layer-8 SAE off-distribution, either to other layers, or to other datasets. As the test layer moves further from train the number of pairwise violations increases, fig. 2A. Similarly, while roughly similar datasets like English wikipedia show no pairwise violations, French or German wikipedia, and code datasets do, even when testing only the pairwise version of eq. (6), fig. 2B. These violations tend to be hierarchical, like a sparse 'french loan word feature' that activates widely on french wikipedia including at all times when a grammatical feature, like "as/like" activates. The analogous code example is a parent encoding stray space in prose or indentations in code, making it a strict superset of another feature that fires within nested code indentations (though on OWT it encodes islands and beaches...).

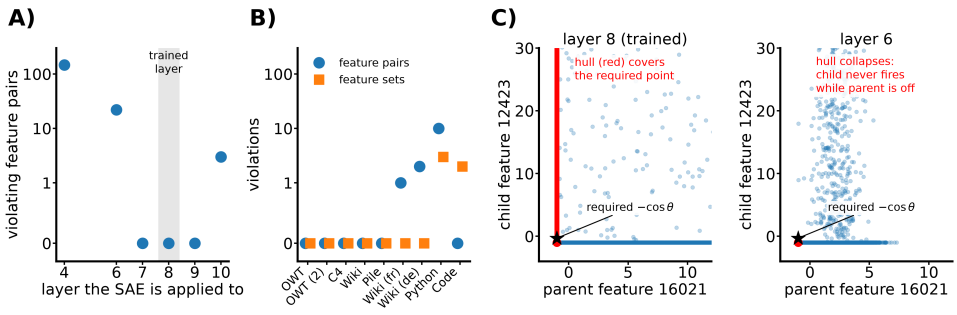


Figure 2: **A)** We apply an SAE trained on layer 8 to other layers and find the number of pairs of features that break eq. (6) increases with distance from layer 8. Similarly, **(B)** when applying an SAE trained on OWT to other datasets the number of violating feature pairs (blue dots) or sets of features (orange squares, we did not test exhaustively) increases with semantic distance from OWT. **C)** An example feature pair that within distribution satisfies the convex hull conditions, no longer does when tested out-of-distribution, as the child never fires while the parent is inactive.

5 Error Residuals

Next, we consider first-order conditions on the feature-residual relationships, these preclude the dictionary from learning concepts that are too similar to patterns it leaves in the residuals.

Constraining Feature-Residual Relationships In app. B.4 we show that the general first-order stability condition, in which we consider all perturbations to $\boldsymbol{Z}$ that preserve nonnegativity, can be rewritten as a constraint on the behaviour of residuals:

$$-\frac{1}{N}\hat{\boldsymbol{w}}_d^T(\bar{\boldsymbol{X}} - \boldsymbol{W}\bar{\boldsymbol{Z}})\delta\boldsymbol{z} = -\frac{1}{N}\hat{\boldsymbol{w}}_d^T\boldsymbol{E}\delta\boldsymbol{z} \in \lambda\text{Convex Hull}(\{\delta\boldsymbol{z}_i | z_d^{[i]} = 0\}) \tag{7}$$

where $\delta\boldsymbol{z}$ is an arbitrary mean-zero response vector and we've defined the residual matrix, $\boldsymbol{E}$. This shows a natural dependence on λ: harsher regularisation means looser constraints on the residuals.

Intuition $\delta\boldsymbol{z}$ is a mean-zero pattern of activity that could be added to neuron d's response. Doing so would change both the reconstruction and regularisor. $\hat{\boldsymbol{w}}_d^T\boldsymbol{E}$ is the residual along neuron d's direction, hence $\frac{1}{N}\hat{\boldsymbol{w}}_d^T\boldsymbol{E}\delta\boldsymbol{z}$ is the correlation between the projected error and the new activity $\delta\boldsymbol{z}$, which determines how much better the reconstruction would be if this perturbation were added. Conversely, adding $\delta\boldsymbol{z}$ to $\boldsymbol{z}$ while preserving non-negativity requires shifting by a bias, which incurs an additional L1 cost whose size is determined by set of values $\delta\boldsymbol{z}$ takes while $\boldsymbol{z}_d$ is inactive. If the gain from reconstruction is smaller than the L1 cost the left hand side will lie within the convex hull on the right hand side, the condition will be satisfied, and the representation will be stable.

Let's break the residual into singular vectors, $\boldsymbol{E} = \sum_i \boldsymbol{u}_i\bar{\boldsymbol{s}}_i^T$ ($||\boldsymbol{u}|| = 1$ without loss of generality), and, considering each $\bar{\boldsymbol{s}}_i$ as a missing feature, derive an interesting necessary condition using $\delta\boldsymbol{z} = \bar{\boldsymbol{s}}$:

$$-\hat{\boldsymbol{w}}_d^T\boldsymbol{u} \in \lambda\text{Convex Hull}\left(\left\{\frac{1}{\sigma_s^2}\bar{s}^{[i]} | z_d^{[i]} = 0\right\}\right), \quad \sigma_s^2 = \frac{1}{N}||\bar{\boldsymbol{s}}||^2 \tag{8}$$

Thus, to be stable, the cosine similarity between the missing concept's direction and $\boldsymbol{w}_d$ must be within the range $\bar{\boldsymbol{s}}$ takes while the neuron is inactive scaled by $\frac{\lambda}{\sigma_s^2}$. We present three implications:

- Similar to our earlier range-independent results, if a feature varies widely while each dictionary element is off, it can be stably left in the residual, fig. 3A.
- Range-dependency destabilises the solution. Hierarchical features are again problematic, if 'labrador' is off whenever 'dog' is, an optimal dictionary containing a 'dog' feature is forced to mop the 'labrador' direction into the 'dog' feature, no matter how orthogonal, fig. 3B.
- Softer hierarchies are destabilised one way. If a sub-feature ('yellow') only activates maximally with a super-feature ('daytime'), otherwise activating weakly then a 'daytime' neuron would be destabilised by the 'yellow' residual, but only if they anti-align, fig. 3C.

We again test eq. (7) on the OWT-trained SAE tested on other datasets by measuring a residual break measure of violation, app. D. We found that the clearest violations again arose from features that activated to whitespace in code datasets, but formatting or byte-level information in OWT, reflecting missing representational structure could usefully have been captured in the code datasets.

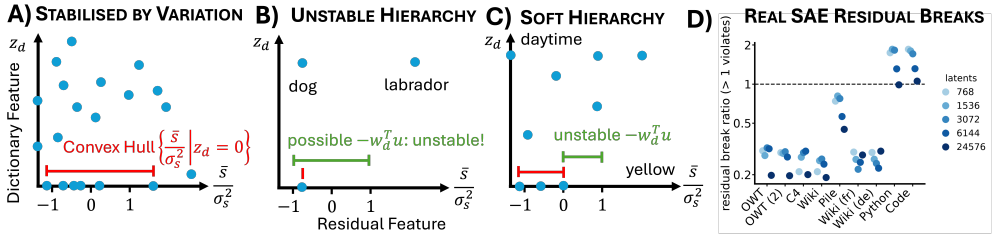


Figure 3: **A)** We plot a dictionary against missing feature; if the missing feature varies widely enough while the dictionary is inactive, it can be stably left in the residuals eq. (7). **B)** Hierarchical features don't survive this check, while **(C)** soft hierarchies fail in one direction. **D)** For OWT-trained SAEs across sizes we find residual breaks in code datasets.

6 Dense Antipodal Feature Pairs

Dense features provide a case study in SAEs behaving strangely when applied to datasets which don't match sparsity assumptions, which we show can be understood simply as a consequence of dictionary learning optimality. Sun et al. (2025) find that SAEs trained on the residual stream of Gemma 2 (Team et al., 2024) learn dense latents that activate on 10 to 50% of datapoints. Further, when using a nonnegative activation function, like JumpReLU or TopK, these latents form antipodal pairs whose dictionary vectors, $\boldsymbol{w}_d$, point in opposite directions. But these pairs disappear when using an activation function that permits both positive and negative latents, such as AbsoluteTopK which thresholds features using the absolute value, rather than a ReLU (Sun et al., 2025).

This result is somewhat intuitive: if forced to code a dense variable into a nonnegative sparse code, one might try breaking the variable into a positive and negative part, something made unnecessary by using AbsoluteTopK. We wondered whether this would always be the case? Intuitively, it seems non-obvious that two antipodal features are sufficient - if there is a small cluster of very high activity along one feature dimension perhaps the cost can be reduced by encoding it separately, fig. 4A?

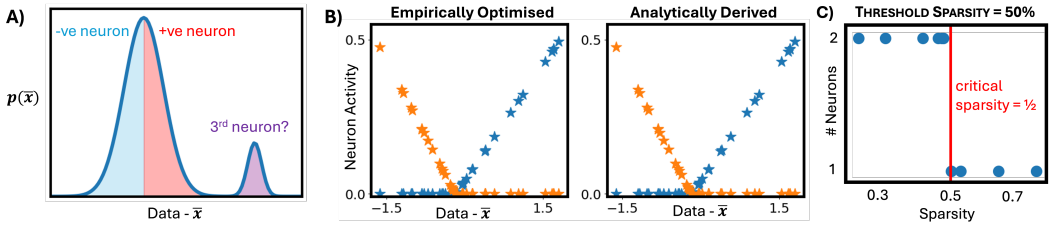


Figure 4: **A)** While splitting a dense variable into positive and negative halves seems intuitive, perhaps outlying clumps should have their own neuron? **(B)** We find this is not the case, an optimal solution to eq. (4) for rank 1 data exists with at most 2 non-zero neurons. This theoretically predicted low-neuron representation, app. E, fits exactly. **C)** Theory predicts that a variable that is more than half sparse should be encoded with only one active neuron: empirics agree.

We find this is not the case. In app. E we study a simple example of a vanilla SAE encoding a single dimension, a scalar datapoint $x^{[i]}$. The sparsity of this variable is defined as the proportion of datapoints for which it equals its minimal value. In this case we can analytically solve the KKT conditions for the optimal SAE, which contains either one neuron if $x^{[i]}$ is sparse, or two when $x^{[i]}$ is dense, and these two neurons have antipodal decoder directions matching both empirics (Sun et al., 2025) and theory on related neural coding problems (Karklin & Simoncelli, 2011; Barrett et al., 2016; Dorrell et al., 2026), fig. 4B. In the small λ limit, we derive the threshold between the one and two neuron regimes: a variable that is active less than half the time will be given one neuron, more and it will be broken into pairs, something we confirm in simulation, fig. 4C. This matches the finding that the dense features never activate more than 50% of the time (Sun et al., 2025). In the following section, we understand this as a special case of a broader phenomenon: ray-clustering.

7 Ray-clustering and the wide convex limit

Finally, we show how dictionary learning optimality can be used to reason about the progressively finer feature splitting observed in SAEs. This feature splitting raises a question: as you increase dictionary size, will splitting continue *ad infinitum*? In short, yes. In the wide convex limit there exists a globally optimal one-dictionary-atom-per-datapoint solution (Bach et al., 2008). For a simple route to this conclusion, consider the original objective, eq. (1) and imagine a solution in which two distinct dictionary atoms (i.e. with different dictionary vectors $\boldsymbol{w}_d$) activate for a single datapoint, fig. 5A. This solution's L1 cost can be improved by instead fitting that datapoint with the same accuracy using a single dictionary element aligned with the datapoint, fig. 5A, leading to the one-latent-per-datapoint.

However, section 6 teaches us that globally minimal solutions can sometimes be reached using much narrower dictionaries, down to 1 neuron. In general what governs how the minimal width globally minimal solution? We show it depends on three things: the regularisation hyperparameter λ, the optimal bias, $\boldsymbol{b}$, and the number of rays from $\boldsymbol{b}$ required to hit every datapoint further than λ from $\boldsymbol{b}$,

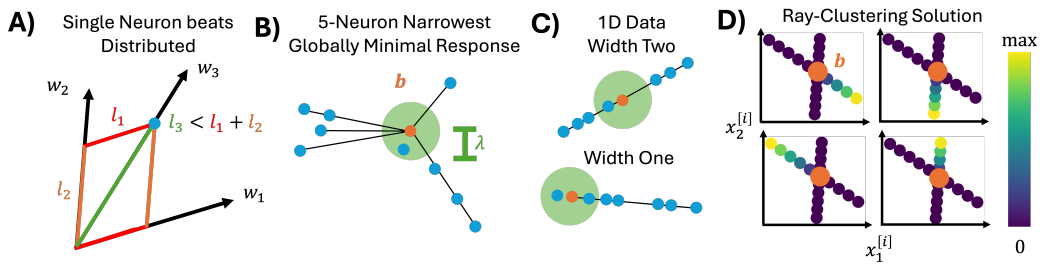


Figure 5: **A)** If a datapoint activates two neurons, and neurons are in abundance, it can be equally well-explained by a third that is aligned and hence uses less activity. **B)** If datapoints share rays from the optimal bias, $\boldsymbol{b}$, it can be explained using only one neuron, narrowing the threshold number of neurons at which the dictionary transitions to the wide globally-minimal solution. Further narrowing comes from data within a λ-ball of $\boldsymbol{b}$ that doesn't have to be encoded at all. **C)** In 1D this becomes the one or two neuron solutions from fig. 4. **(D)** We train a 30 neuron dictionary on this 2D cross dataset. We find only the four pictured neurons activate. The geometric median lands in the middle of the cross, and an optimal dictionary uses only four active neurons to explain the four rays of the data.

fig. 5B. In app. F, we derive $\boldsymbol{b}$. Under perfect reconstruction ($\lambda \to 0$) it is the geometric median, the multidimensional generalisation of the median:

$$\boldsymbol{b} = \arg\min_{\boldsymbol{b}} \sum_i ||\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{b}||_2 \tag{9}$$

Under imperfect reconstruction $\boldsymbol{b}$ is a slight generalisation, app. F. When $\lambda = 0$ the number of rays required to hit every datapoint from $\boldsymbol{b}$ then determines the width of the narrowest global minima. This explains section 6: in one dimension $\boldsymbol{b}$ is simply the median; if more than 50% of the data lies on an extreme point the median lies on an extreme point and only one ray is required, else, two are, fig. 5C.

At finite λ we derive a second-order condition on whether an additional neuron can reduce the loss:

$$\max_i ||\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{W}\bar{\boldsymbol{z}}^{[i]}||_2 \leq \lambda \tag{10}$$

i.e. a solution is stable to new neurons if the residuals for all datapoints are below λ. If a locally optimal representation passes this test, it is a globally optimal solution, since it can be padded to with sufficient dead neurons to make the problem convex. This introduces a λ-buffer region of data around $\boldsymbol{b}$ that don't activate any neurons, reducing the width of the narrowest global minima still further.

8 Discussion

In this paper we studied how optimality in a dictionary learning problem, eq. (1), structures learnt dictionaries, and we used this to puzzle over representational properties of SAEs - including hierarchical splitting & absorption, the structure of residuals, dense antipodal features, and the neuron-per-datapoint-limit. This logic is pleasingly simple, unifying diverse behaviours in a single framework.

Our results have clear limits. The conditions are necessary rather than sufficient, are derived for vanilla dictionary learning rather than specific encoders, and are tested only pairwise and on subsets of features. Two directions seem particularly useful. First, applying similar approaches to encoder architecture could link the choice of SAE to its representational assumptions, as we have done here for dictionary learning. Second, it remains unclear whether the linear representation hypothesis and feature superposition accurately reflect how neural networks represent information. Tools that separate what is imposed by the SAE from what is present in the network are a prerequisite for answering this.

Most broadly, our observations are filtered through the tools we use and the assumptions they necessarily carry. Dictionary learning is at least clear about its assumptions, and this paper uses these assumptions to interpret observed phenomena as products of this filtering. Practically, the same conditions offer a cheap diagnostic for when an SAE's dictionary no longer fits the data it is applied to. More ambitiously, by emphasising the outsized role these assumptions play, we hope to motivate the development of alternative interpretability tools with more realistic representational assumptions.

AI USE STATEMENT

In this work we used generative AI tools for checking mathematical arguments, helping discover relevant papers and surveying the literature, poking holes in the paper, and coding up the experimental work. While AI has checked every line of the maths, it only wrote those that appear in app. C. The authors are to blame for most of the logic barring one very helpful case, where Claude recognised the geometric median and provided links to its speedy computation, a concept we had never heard of before. Finally, we did not write the code that was used to generate the non-toy experiments. Needless to say, we take responsibility for the final content of this work.

REPRODUCIBILITY

All mathematical arguments are included in the appendix. Code to run all experiments can be found at the following anonymised git repo: `https://anonymous.4open.science/r/SAE_Identifiability-3243/README.md`.

REFERENCES

Kurt M Anstreicher. On convex relaxations for quadratically constrained quadratic programming. *Mathematical programming*, 136(2):233–251, 2012.

Dana Arad, Aaron Mueller, and Yonatan Belinkov. Saes are good for steering–if you select the right features. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10252–10270, 2025.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495, 2018.

Joseph J Atick and A Norman Redlich. Towards a theory of early visual processing. *Neural computation*, 2(3):308–320, 1990.

Francis Bach, Julien Mairal, and Jean Ponce. Convex sparse matrix factorizations. *arXiv preprint arXiv:0812.1869*, 2008.

David GT Barrett, Sophie Deneve, and Christian K Machens. Optimal compensation for neuron loss. *Elife*, 5:e12454, 2016.

Abraham Berman and Naomi Shaked-Monderer. *Completely positive matrices*. World Scientific, 2003.

Joseph Bloom. Open source sparse autoencoders for all residual stream layers of gpt2 small. `https://www.alignmentforum.org/posts/f9EgfLSurAiqRJySD/open-source-sparse-autoencoders-for-all-residual-stream`, 2024.

Joseph Bloom, Curt Tigges, Anthony Duong, and David Chanin. Saelens. `https://github.com/decoderesearch/SAELens`, 2024.

Stephen Boyd and Lieven Vandenberghe. Convex optimization. *Cambridge UP*, 2004.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.

Bart Bussmann, Patrick Leask, and Neel Nanda. Batchtopk sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. *arXiv preprint arXiv:2503.17547*, 2025.

Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, 52(12):5406–5425, 2006.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *arXiv preprint arXiv:2409.14507*, 2024.

Jeremy E Cohen and Nicolas Gillis. Identifiability of complete dictionary learning. *SIAM Journal on Mathematics of Data Science*, 1(3):518–536, 2019.

Valérie Costa, Thomas Fel, Ekdeep Singh Lubana, Bahareh Tolooshams, and Demba Ba. From flat to hierarchical: Extracting sparse representations with matching pursuit. *arXiv preprint arXiv:2506.03093*, 2025.

Jingyi Cui, Qi Zhang, Yifei Wang, and Yisen Wang. On the limits of sparse autoencoders: A theoretical framework and reweighted remedy. In *The Fourteenth International Conference on Learning Representations*, 2026.

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

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

Peter JC Dickinson and Luuk Gijben. On the computational complexity of membership problems for the completely positive cone and its dual. *Computational optimization and applications*, 57(2): 403–415, 2014.

David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? *Advances in neural information processing systems*, 16, 2003.

David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization. *Proceedings of the National Academy of Sciences*, 100(5): 2197–2202, 2003.

William Dorrell, Kyle Hsu, Luke Hollingsworth, Jin Hwa Lee, Jiajun Wu, Chelsea Finn, Peter E Latham, Timothy Edward John Behrens, and James CR Whittington. Range, not independence, drives modularity in biologically inspired representations. In *The Thirteenth International Conference on Learning Representations*, 2025.

William Dorrell, Peter E Latham, and James Whittington. Convex efficient coding. *arXiv preprint arXiv:2601.10482*, 2026.

Bogdan Dumitrescu and Paul Irofti. *Dictionary learning algorithms and applications*. Springer, 2018.

Esin Durmus, Alex Tamkin, Jack Clark, Jerry Wei, Jonathan Marcus, Joshua Batson, Kunal Handa, Liane Lovitt, Meg Tong, Miles McCain, et al. Evaluating feature steering: A case study in mitigating social biases. *Anthropic Research*, 2024.

Michael Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Thomas Fel, Ekdeep Singh Lubana, Jacob S Prince, Matthew Kowal, Victor Boutin, Isabel Papadimitriou, Binxu Wang, Martin Wattenberg, Demba Ba, and Talia Konkle. Archetypal sae: Adaptive and stable dictionary learning for concept extraction in large vision models. *arXiv preprint arXiv:2502.12892*, 2025.

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

Quan Geng and John Wright. On the local correctness of l_1-minimization for dictionary learning. In *2014 IEEE International Symposium on Information Theory*, pp. 3180–3184. IEEE, 2014.

Nicolas Gillis. *Nonnegative matrix factorization*. SIAM, 2020.

Rémi Gribonval and Karin Schnass. Dictionary identification—sparse matrix-factorization via l_1-minimization. *IEEE Transactions on Information Theory*, 56(7):3523–3539, 2010.

Rémi Gribonval, Rodolphe Jenatton, and Francis Bach. Sparse and spurious: dictionary learning with noise and outliers. *IEEE Transactions on Information Theory*, 61(11):6298–6319, 2015.

Elaine T Hale, Wotao Yin, and Yin Zhang. Fixed-point continuation for \ell_1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008.

Christopher J Hillar and Friedrich T Sommer. When can dictionary learning uniquely recover sparse data from subsamples? *IEEE Transactions on Information Theory*, 61(11):6290–6297, 2015.

Sai Sumedh R Hindupur, Ekdeep Singh Lubana, Thomas Fel, and Demba Ba. Projecting assumptions: The duality between sparse autoencoders and concept geometry. *arXiv preprint arXiv:2503.01822*, 2025.

Jingzhou Hu and Kejun Huang. Global identifiability of l1-based dictionary learning via matrix volume optimization. *Advances in Neural Information Processing Systems*, 36:36165–36186, 2023.

Robert Israel. Is the trace of inverse matrix convex?, 2013. URL `https://math.stackexchange.com/questions/297635/is-the-trace-of-inverse-matrix-convex`.

Yan Karklin and Eero Simoncelli. Efficient coding of natural images with a population of noisy linear-nonlinear neurons. *Advances in neural information processing systems*, 24, 2011.

Jean Bernard Lasserre. *Moments, positive polynomials and their applications*, volume 1. World Scientific, 2009.

Simon Laughlin. A simple coding procedure enhances a neuron's information capacity. *Zeitschrift für Naturforschung c*, 36:910–912, 1981.

Patrick Leask, Bart Bussmann, Michael Pearce, Joseph Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis. *arXiv preprint arXiv:2502.04878*, 2025.

Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69, 2008.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pp. 746–751, 2013.

Theodore Samuel Motzkin. Beiträge zur theorie der linearen ungleichungen. *University of Basel (Doctoral Thesis)*, 1936.

Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

Bruno A Olshausen and David J Field. Sparse coding of sensory inputs. *Current opinion in neurobiology*, 14(4):481–487, 2004.

Charles O'Neill, Alim Gumran, and David Klindt. Compute optimal inference and provable amortisation gap in sparse autoencoders. *arXiv preprint arXiv:2411.13117*, 2024.

Cengiz Pehlevan and Dmitri B Chklovskii. A hebbian/anti-hebbian network derived from online non-negative matrix factorization can cluster and discover sparse features. In *2014 48th Asilomar Conference on Signals, Systems and Computers*, pp. 769–775. IEEE, 2014.

Javier Pena, Juan C Vera, and Luis F Zuluaga. Completely positive reformulations for polynomial optimization. *Mathematical Programming*, 151(2):405–431, 2015.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024.

Karsten Roth, Mark Ibrahim, Zeynep Akata, Pascal Vincent, and Diane Bouchacourt. Disentanglement of correlated factors via hausdorff factorized support. In *The Eleventh International Conference on Learning Representations*, 2023.

Christopher J Rozell, Don H Johnson, Richard G Baraniuk, and Bruno A Olshausen. Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, 20(10):2526–2563, 2008.

Srivatsun Sadagopan and Xiaoqin Wang. Nonlinear spectrotemporal interactions underlying selectivity for complex sounds in auditory cortex. *Journal of neuroscience*, 29(36):11192–11202, 2009.

Karin Schnass. Local identification of overcomplete dictionaries. *The Journal of Machine Learning Research*, 16(1):1211–1242, 2015.

Anirvan Sengupta, Cengiz Pehlevan, Mariano Tepper, Alexander Genkin, and Dmitri Chklovskii. Manifold-tiling localized receptive fields are optimal in similarity-preserving neural networks. *Advances in neural information processing systems*, 31, 2018.

Naomi Shaked-Monderer, Immanuel M Bomze, Florian Jarre, and Werner Schachinger. On the cp-rank and minimal cp factorizations of a completely positive matrix. *SIAM Journal on Matrix Analysis and Applications*, 34(2):355–368, 2013.

Naum Z Shor. Class of global minimum bounds of polynomial functions. *Cybernetics*, 23(6):731–734, 1987.

Eero P Simoncelli and Bruno A Olshausen. Natural image statistics and neural representation. *Annual review of neuroscience*, 24(1):1193–1216, 2001.

Xiaoqing Sun, Alessandro Stolfo, Joshua Engels, Ben Wu, Senthooran Rajamanoharan, Mrinmaya Sachan, and Max Tegmark. Dense sae latents are features, not bugs. *arXiv preprint arXiv:2506.15679*, 2025.

Yiming Tang, Harshvardhan Saini, Zhaoqian Yao, Yizhen Liao, Qianxiao Li, Mengnan Du, and Dianbo Liu. On the theoretical foundation of sparse dictionary learning in mechanistic interpretability. *arXiv preprint arXiv:2512.05534*, 2025.

Gokcan Tatli and Alper T Erdogan. Polytopic matrix factorization: Determinant maximization based criterion and identifiability. *IEEE Transactions on Signal Processing*, 69:5431–5447, 2021.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

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.

Demian Till. Do sparse autoencoders find "true features", 2024. URL https://www.lesswrong.com/posts/QoR8noAB3Mp2KBA4B/do-sparse-autoencoders-find-true-features.

Yu Wang, Siqi Wu, and Bin Yu. Unique sharp local minimum in l1-minimization complete dictionary learning. *Journal of Machine Learning Research*, 21(63):1–52, 2020.

Ben DB Willmore and Andrew J King. Auditory cortex: representation through sparsification? *Current Biology*, 19(24):R1123–R1125, 2009.

John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2): 210–227, 2008.

Siqi Wu and Bin Yu. Local identifiability of l_1-minimization dictionary learning: a sufficient and almost necessary condition. *Journal of Machine Learning Research*, 18(168):1–56, 2018.

A Dictionary Learning Reformulations & Convexity

A.1 Problem Statement

We consider a standard formulation of dictionary learning. Given each of N datapoints, $\boldsymbol{x}^{[i]} \in \mathbb{R}^{d_x}$, we seek a sparse representation, $\boldsymbol{z}^{[i]} \in \mathbb{R}^{d_z}$, from which we can linearly reconstruct the data:

$$\boldsymbol{x}^{[i]} \approx \boldsymbol{W}\boldsymbol{z}^{[i]} + \boldsymbol{b} \tag{11}$$

where $\boldsymbol{b}$ is a bias and $\boldsymbol{W}$ is the dictionary matrix. For terminology, we will refer to each element of $\boldsymbol{z}^{[i]}$ as a neuron or feature, and each column of the dictionary matrix, $\boldsymbol{w}_d$, as the feature's direction or dictionary element. To enforce the dual goals of accurate fitting of data, eq. (11), and sparsity, we use the following optimisation problem, eq. (1) in the main text:

$$\min_{\boldsymbol{W},\{\boldsymbol{z}^{[i]}\}_{i=1}^N,\boldsymbol{b}} \mathbb{E}_i\left[||\boldsymbol{W}\boldsymbol{z}^{[i]}+\boldsymbol{b}-\boldsymbol{x}^{[i]}||_2^2+2\lambda||\boldsymbol{z}^{[i]}||_1\right], \qquad \text{subject to} \quad \boldsymbol{z}^{[i]} \geq 0, \quad ||\boldsymbol{w}_d||_2 = 1 \tag{12}$$

We make the standard substitution of enforcing sparsity using the continuous (and convex) L1 norm, rather than L0, for mathematical practicality, and we define λ with a factor of 2 for later convenience. The unit-norm constraint on the dictionary columns is introduced to circumvent the scaling degeneracy between representations, $\boldsymbol{z}^{[i]}$, and dictionary elements, $\boldsymbol{W}$. Finally, the non-negativity of the codes reflects a common intuition that each dictionary element should contribute only positively to the representation.

Bricken et al. (2023) study exactly this optimisation problem with one important difference. Rather than inferring $\boldsymbol{z}^{[i]}$ by solving eq. (12), they 'amortise the inference' of $\boldsymbol{z}^{[i]}$ via an encoder from $\boldsymbol{x}^{[i]}$:

$$\boldsymbol{z}^{[i]} = \text{ReLU}(\boldsymbol{W}_{\text{enc}}\boldsymbol{x}^{[i]} + \boldsymbol{b}_{\text{enc}}) \tag{13}$$

This combination, eq. (12) and eq. (13), represents the 'vanilla sparse autoencoder' (SAE) problem.

The Affine+ReLU encoder structure is important, placing constraints on the learnable SAE representations. However, we will ignore it, instead studying solutions to the original dictionary learning problem eq. (12). In some cases this idealised model and vanilla SAEs are identical, for example, when the dictionary elements are orthogonal app. B.2, but this is not generally true. Instead this choice reflects an idealisation. There's a zoo of possible SAE architectures and the way architecture constrains learnt features has been studied(Hindupur et al., 2025). Instead, we ask how the dictionary learning objective alone shapes optimal representations. This is not an unfeasible idealisation, indeed algorithms exist to recover these optimal solutions, requiring only the addition of recurrent weights to the standard Affine+ReLU SAE app. B.2.1, and these have been used within the standard SAE pipeline (O'Neill et al., 2024). We therefore view vanilla dictionary learning as a good 'model organism' to help us explain representational patterns across SAE architectures.

A.2 Reformulations Using Scaling Symmetry

In this section we will use scaling symmetry to equate three versions of dictionary learning.

Consider the following optimisation problem, eq. (3) in the main text, that initially appears quite different to the original dictionary learning problem eq. (12):

$$\min_{\boldsymbol{W},\{\boldsymbol{z}^{[i]}\}_{i=1}^N,\boldsymbol{b}} \mathbb{E}_i\left[||\boldsymbol{W}\boldsymbol{z}^{[i]}+\boldsymbol{b}-\boldsymbol{x}^{[i]}||_2^2\right] + \frac{\lambda}{N}\sum_{d=1}^{d_z}(||\boldsymbol{w}_d||_2^2+||\boldsymbol{z}_d||_1^2), \qquad \text{subject to} \quad \boldsymbol{z}^{[i]} \geq 0 \tag{14}$$

We develop it using an argument from Bach et al. (2008). The reconstruction error is invariant to rescaling each neuron's activities and weights with a neuron-dependent scalar:

$$\{\boldsymbol{w}_d, \boldsymbol{z}_d\}_{d=1}^{d_z} \to \{\sigma_d \boldsymbol{w}_d, \frac{1}{\sigma_d}\boldsymbol{z}_d\}_{d=1}^{d_z} \tag{15}$$

That means we can optimise each σ_d to minimise only the regularisation term. Performing this optimisation, one finds the optimal value is:

$$(\sigma_d^*)^2 = \frac{||\boldsymbol{z}_d||_1}{||\boldsymbol{w}_d||_2} \tag{16}$$

Inserting this value into eq. (14) we recover eq. (2) from the main text, the dictionary learning problem with unconstrained norm:

$$\min_{\boldsymbol{W},\{\boldsymbol{z}^{[i]}\}_{i=1}^{N},\boldsymbol{b}} \mathbb{E}_i\Big[||\boldsymbol{W}\boldsymbol{z}^{[i]}+\boldsymbol{b}-\boldsymbol{x}^{[i]}||_2^2\Big]+\frac{2\lambda}{N}\sum_{d=1}^{d_z}||\boldsymbol{w}_d||_2||\boldsymbol{z}_d||_1, \qquad \text{subject to} \quad \boldsymbol{z}^{[i]}\geq 0 \tag{17}$$

So, since any optimal solution will eventually settle to the optimal scaling σ_d^*, solving the original problem, eq. (14), is equivalent to solving eq. (17) then rescaling optimally. Finally, however, eq. (17) is now completely invariant to neuron-wise rescalings, so we can choose, for example, to only consider solutions in which $||\boldsymbol{w}_d||_2 = 1$, recovering the original formulation, eq. (12). One thing this illustrates is that the SAE improvements reported by Templeton et al. (2024) upon moving from eq. (12) to eq. (17) must reflect better optimisations, since the properties of the optimal solutions are not meaningfully changed.

A.3 Reformulation by Optimising over Decoder

Now we will optimise over the decoder to define a problem only in terms of the sparse representation, again following the same path as Bach et al. (2008).

First, we introduce a few further matrices, $\boldsymbol{Z} \in \mathbb{R}^{d_z \times N}$ and $\boldsymbol{X} \in \mathbb{R}^{d_x \times N}$, the stacked SAE and network representation matrices, and their row-mean subtracted partners, $\bar{\boldsymbol{Z}}$ and $\bar{\boldsymbol{X}}$:

$$\bar{\boldsymbol{Z}} = \boldsymbol{Z} - \frac{1}{N}\boldsymbol{Z}\boldsymbol{1}\boldsymbol{1}^T \tag{18}$$

We can optimise eq. (14) directly over the bias $\boldsymbol{b}$. Since it is unconstrained, it can mop up any errors in reconstruction along the mean direction, letting us write the optimisation problem as:

$$\min_{\boldsymbol{Z},\boldsymbol{W}} \frac{1}{N}||\boldsymbol{W}\bar{\boldsymbol{Z}}-\bar{\boldsymbol{X}}||_F^2+\frac{\lambda}{N}\big(\sum_{d=1}^{d_z}||\boldsymbol{z}_d||_1^2+||\boldsymbol{w}_d||_2^2\big), \quad \text{subject to} \quad \boldsymbol{Z}\geq 0 \tag{19}$$

We see that we can now optimise directly over $\boldsymbol{W}$, finding:

$$\boldsymbol{W}^* = \bar{\boldsymbol{X}}\bar{\boldsymbol{Z}}^T(\lambda\mathbb{1}+\bar{\boldsymbol{Z}}\bar{\boldsymbol{Z}}^T)^{-1} = \bar{\boldsymbol{X}}(\lambda\mathbb{1}+\bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{Z}}^T \tag{20}$$

Using this and dropping a shared $\frac{\lambda}{N}$ factor we find the following optimisation problem in $\boldsymbol{Z}$:

$$\min_{\boldsymbol{Z}} \mathrm{Tr}[(\lambda\mathbb{1}+\bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}]+\boldsymbol{1}^T\boldsymbol{Z}^T\boldsymbol{Z}\boldsymbol{1}, \quad \text{subject to} \quad \boldsymbol{Z}\geq 0 \tag{21}$$

Where we have rewritten the L1 loss penalty into a convenient matrix form. This is eq. (4) from the main text, which we will use to derive our key results.

A.4 Convexity in the Large Dictionary Limit

Here we show that we can rewrite eq. (21) in terms of the representational similarity, defined as:

$$Q_{i,j} = \boldsymbol{z}^{[i],T}\boldsymbol{z}^{[j]} \tag{22}$$

Surprisingly, and following existing work in dictionary learning (Bach et al., 2008) and theoretical neuroscience (Sengupta et al., 2018; Dorrell et al., 2026), when we do this we find that the problem is convex when there are enough neurons. This is not a practical setting, but will prove useful to us in clarifying the structure of the loss landscape, and deriving limiting behaviours of SAEs.

Activity Loss: Using the non-negativity of $\boldsymbol{z}_d$, we can rewrite the activity loss:

$$\sum_{d=1}^{d_z}||\boldsymbol{z}_d||_1^2 = \sum_{d=1}^{d_z}(\sum_{i=1}^{N}z_d^{[i]})^2 = \sum_{d,i,j=1}^{d_z,N,N}z_d^{[i]}z_d^{[j]} = \sum_{i,j=1}^{N,N}(\boldsymbol{z}^{[i]})^T\boldsymbol{z}^{[j]} = \sum_{i,j=1}^{N,N}Q_{ij} = \boldsymbol{1}^T\boldsymbol{Q}\boldsymbol{1} \tag{23}$$

which is a linear, and therefore convex, function of $\boldsymbol{Q}$.

Nonnegativity Constraint The set of dot product matrices with unconstrained ranks formed from vectors that are themselves nonnegative ($\boldsymbol{Q} = \boldsymbol{Z}^T\boldsymbol{Z}$ for $\boldsymbol{Z}\geq 0$) form a convex set called the set of completely positive matrices (Berman & Shaked-Monderer, 2003). However, $\boldsymbol{Q}$ must additionally be

constrained to have a completely-positive rank lower than d_z, meaning that there exists a nonnegative matrix $\boldsymbol{A} \geq 0$ with at most d_z rows such that $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{Q}$. This is a non-convex constraint. The cp-rank of an $N \times N$ matrix is at most $N(N+1)/2$ (Shaked-Monderer et al., 2013), so when $d_z \geq N(N+1)/2$ we can drop the cp-rank constraint making the whole problem convex.

Weight and Reconstruction Loss The weight and reconstruction loss can be written in terms of $\boldsymbol{Q}$:

$$\min_{\boldsymbol{Z}} \mathrm{Tr}[(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}] = \min_{\boldsymbol{Q}} \mathrm{Tr}[(\lambda\mathbb{1} + \bar{\boldsymbol{Q}})^{-1}\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}] \tag{24}$$

For completeness we prove convexity. Following Dorrell et al. (2026), we modify the proof technique of maths stack exchange user Robert Israel (Israel, 2013). A function, $\mathcal{L}$, of positive semi-definite matrix, $\bar{\boldsymbol{Q}}$ is convex if for $\bar{\boldsymbol{Q}}(t) = \bar{\boldsymbol{Q}} + t\boldsymbol{S}$ where $\boldsymbol{S}$ is a symmetric matrix:

$$\left.\frac{d^2\mathcal{L}(\bar{\boldsymbol{Q}}(t))}{dt^2}\right|_{t=0} \geq 0 \tag{25}$$

Using the fact that:

$$(\boldsymbol{A} + t\boldsymbol{S})^{-1} = \boldsymbol{A}^{-1} - t\boldsymbol{A}^{-1}\boldsymbol{S}\boldsymbol{A}^{-1} + t^2\boldsymbol{A}^{-1}\boldsymbol{S}\boldsymbol{A}^{-1}\boldsymbol{S}\boldsymbol{A}^{-1} \tag{26}$$

with the identification $\boldsymbol{A} = \lambda\mathbb{1} + \bar{\boldsymbol{Q}}$ we find:

$$\frac{1}{2}\left.\frac{d^2\mathcal{L}(\bar{\boldsymbol{Q}}(t))}{dt^2}\right|_{t=0} = \mathrm{Tr}[\boldsymbol{A}^{-1}\boldsymbol{S}\boldsymbol{A}^{-1}\boldsymbol{S}\boldsymbol{A}^{-1}\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}] \tag{27}$$

Finally, writing $\boldsymbol{C} = \boldsymbol{S}\boldsymbol{A}^{-1}\bar{\boldsymbol{X}}^T$ this is:

$$\frac{1}{2}\left.\frac{d^2\mathcal{L}(\bar{\boldsymbol{Q}}(t))}{dt^2}\right|_{t=0} = \mathrm{Tr}[\boldsymbol{C}^T\boldsymbol{A}^{-1}\boldsymbol{C}] \tag{28}$$

But we know $\boldsymbol{A}^{-1} = (\lambda\mathbb{1} + \bar{\boldsymbol{Q}})^{-1}$ is positive definite, hence this quantity is nonnegative and the loss is convex.

Finally, this is a function on demeaned representational similarity matrices. Changing the mean changes $\boldsymbol{Q}$, but doesn't change the weight loss, so as a function of the loss is, first, a demeaning projection (it is easy to check this is a projection), then the loss:

$$P : \boldsymbol{Q} \to \boldsymbol{Q} - \frac{1}{N}\mathbf{1}\mathbf{1}^{\mathbf{T}}\boldsymbol{Q} - \frac{1}{N}\boldsymbol{Q}\mathbf{1}\mathbf{1}^{\mathbf{T}} + \frac{1}{N^2}\mathbf{1}\mathbf{1}^{\mathbf{T}}(\mathbf{1}^T\boldsymbol{Q}\mathbf{1}) \tag{29}$$

$$\text{recon + weight loss} \propto \mathrm{Tr}[(\lambda\mathbb{1} + P(\boldsymbol{Q}))^{-1}\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}] \tag{30}$$

Our previous convexity results then apply to the full similarity matrix.

B LOCAL OPTIMALITY CONDITIONS

We will now derive constraints that $\boldsymbol{Z}$ must satisfy to be locally optimal. To do this we will perturb the representation and find conditions under which the first-order change to the loss is nonnegative. We will consider three types of perturbation, one completely general, and two that probe neuron-neuron and neuron-residual relationships.

B.1 PREAMBLE

To begin, we want our perturbation to preserve nonnegativity, and we want to ignore an 'uninteresting' kind of suboptimality. If any $\boldsymbol{z}_d$ has a minimum value greater than zero the loss can be reduced by shifting this neuron's activity down. We assume this has been done so $\min \boldsymbol{z}_d = 0$. To remove this possibility we parametrise the representation and the perturbation directly in terms of its mean subtracted version, then add the correct offset to preserve nonnegativity:

$$\boldsymbol{Z} \to \bar{\boldsymbol{Z}} + \Delta - \min_{\text{row-wise}}[\bar{\boldsymbol{Z}} + \Delta]\mathbf{1}^{\mathbf{T}} \qquad \bar{\boldsymbol{Z}} \to \bar{\boldsymbol{Z}} + \Delta \tag{31}$$

Where $\Delta \in \mathbb{R}^{d_z \times N}$ is a row-mean-zero perturbation matrix.

The first-order perturbation to the weight-and-reconstruction term is then:

$$-2\,\mathrm{Tr}[(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{Z}}^T\Delta(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}] \tag{32}$$

While the activity loss depends on the minimal de-meaned response of each neuron:

$$\mathbf{1}^T\boldsymbol{Z}^T\boldsymbol{Z}\mathbf{1} = N^2\sum_d(\min[\bar{\boldsymbol{z}}_d + \Delta_d])^2 \tag{33}$$

To first order, the change in minima for each neuron is:

$$(\min[\bar{\boldsymbol{z}}_d+\Delta_d])^2-(\min[\bar{\boldsymbol{z}}_d])^2 = (\min[\bar{\boldsymbol{z}}_d]+\min_{i\in\mathcal{A}_d}[\Delta_d^{[i]}])^2-(\min[\bar{\boldsymbol{z}}_d])^2 = 2\min[\bar{\boldsymbol{z}}_d]\min_{i\in\mathcal{A}_d}[\Delta_d^{[i]}] \tag{34}$$

where $\mathcal{A}_d$ is the set of datapoints at which $z_d^{[i]} = 0$, the inactive set.

We introduce one further notational convenience, we denote with m_d the L1 norm of the dth neuron:

$$m_d = ||\boldsymbol{z}_d||_1 = \mathbf{1}^T\boldsymbol{z}_d = N|\min[\bar{\boldsymbol{z}}_d]| \tag{35}$$

By the scaling arguments presented in app. A.2, this is also equal to the L2 norm of the dth column of the dictionary: $||\boldsymbol{w}_d||_2 = m_d$.

Hence, half the total first order change in the loss is:

$$\frac{1}{2}\Delta\mathcal{L} = -\sum_d m_d N\min_{i\in\mathcal{A}_d}[\Delta_d^{[i]}] - \mathrm{Tr}[(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{Z}}^T\Delta(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}] \tag{36}$$

Since the weight/reconstruction loss is differentiable and the activity penalty breaks down by neuron, we can consider a first-order perturbation to only one neuron (i.e. Δ has only one non-zero row). If the loss is unstable there must be a neuron-perturbation that would reduce the loss. Hence, we may instead consider the single neuron perturbation:

$$\frac{1}{2}\Delta\mathcal{L}_d = -m_d N\min_{i\in\mathcal{A}_d}[\Delta_d^{[i]}] - \Delta_d^T(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\boldsymbol{z}_d \tag{37}$$

For writing convenience, using the definition of the optimal dictionary matrix, $\boldsymbol{W}^*(\boldsymbol{Z}) = \bar{\boldsymbol{X}}(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{Z}}^T$, we can identify the dth dictionary element, allowing us to clean this up slightly:

$$\frac{1}{2}\Delta\mathcal{L}_d = -m_d N\min_{i\in\mathcal{A}_d}[\Delta_d^{[i]}] - \Delta_d^T(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{X}}^T\boldsymbol{w}_d \tag{38}$$

B.2 A General First-Order Perturbation

If the change in loss, eq. (38), is nonnegative for all mean-zero Δ_d, $\boldsymbol{Z}$ is first-order stable; if we can find one Δ_d that makes this negative it is first-order unstable. Let's see if such an instability inducing Δ_d exists.

Writing a shorthand for a scaling of the gradient of the reconstruction-and-weight loss:

$$\boldsymbol{g}_d = \frac{1}{Nm_d}(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} \bar{\boldsymbol{X}}^T \boldsymbol{w}_d \tag{39}$$

Our condition for instability is:

$$\min_{i \in \mathcal{A}_d} [\Delta_d^{[i]}] > -\Delta_d^T \boldsymbol{g}_d \tag{40}$$

This condition can be equivalently expanded as:

$$\Delta_d^{[i]} > -\Delta_d^T \boldsymbol{g}_d \quad \forall i \in \mathcal{A}_d \tag{41}$$

This is now a set of $|\mathcal{A}_d|$ linear conditions that we can stack as:

$$(\mathbb{1}_{\mathcal{A}_d} + \mathbf{1}\boldsymbol{g}_d^T)\Delta_d > 0 \qquad \mathbf{1}^T \Delta_d = 0 \tag{42}$$

Where $\mathbb{1}_{\mathcal{A}_d} \in \mathcal{R}^{|\mathcal{A}_d| \times N}$ is the matrix that extracts only the elements of Δ_d within $\mathcal{A}_d$, and the last constraint enforces Δ_d mean-zero.

This is a feasibility problem, if we can find a Δ_d that satisfies these constraints $\boldsymbol{Z}$ is not a local minima. We can switch to the dual feasible problem (Boyd & Vandenberghe, 2004) and derive a dual certificate. This is another feasibility problem, only one of which has a solution. In particular, we will use Motzkin's Transposition Theorem (Motzkin, 1936), which tells us a dual certificate is that there exists a solution to:

$$(\mathbb{1}_{\mathcal{A}_d}^T + \boldsymbol{g}_d \mathbf{1}^T)\boldsymbol{y} + \mathbf{1}\mu = 0 \qquad \boldsymbol{y} \geq 0, \qquad \boldsymbol{y} \in \mathbb{R}^{|\mathcal{A}_d|}, \mu \in \mathbb{R} \tag{43}$$

We can project these expressions onto the $\mathbf{1}$ vector and, using the fact that $\boldsymbol{g}_d$ is mean-zero, find:

$$\mu = -\frac{1}{N}\sum_j y_j = -\frac{||\boldsymbol{y}||_1}{N} \tag{44}$$

Using this we get the following constraints on $\boldsymbol{g}_d$:

$$g_d^{[i]} = \begin{cases} \frac{1}{N} & i \notin \mathcal{A}_d \\ \frac{1}{N} - \frac{y_i}{||\boldsymbol{y}||_1} & i \in \mathcal{A}_d \end{cases} \tag{45}$$

This is an interesting set of conditions, telling us that, for $\boldsymbol{Z}$ to be stable, the vector $\boldsymbol{g}_d$ must be constant on the active set of neuron d and smaller on the inactive set, $\mathcal{A}_d$. We can simplify by assuming, wlog, that $||\boldsymbol{y}||_1 = 1$ (since this just corresponds to a positive rescaling of $\boldsymbol{y}$ and μ, something permitted while maintaining the same dual certificate, eq. (43)). We will show that our more specific constraints can be understood as projections of this condition.

We can rewrite it in one more clean fashion, using the definition of $\boldsymbol{g}_d$:

$$m_d N \boldsymbol{g}_d = (\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} \bar{\boldsymbol{X}}^T \boldsymbol{w}_d = m_d \mathbf{1} - m_d N \boldsymbol{y} \tag{46}$$

where, in an abuse of notation, we've used $\boldsymbol{y}$ to denote both the original $|\mathcal{A}_d|$-dimensional vector, and its N dimensional analogue that is padded with zeros for all indices not in $\mathcal{A}_d$. Now using the Woodbury Matrix Inversion formula:

$$(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} = \frac{1}{\lambda}\mathbb{1} - \frac{1}{\lambda}\bar{\boldsymbol{Z}}^T(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}\bar{\boldsymbol{Z}}^T)^{-1}\bar{\boldsymbol{Z}} \tag{47}$$

Plugging this into the expression and identifying the factors of $\boldsymbol{W} = \bar{\boldsymbol{X}}\bar{\boldsymbol{Z}}^T(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}\bar{\boldsymbol{Z}}^T)^{-1}$:

$$\bar{\boldsymbol{X}}^T \hat{\boldsymbol{w}}_d - \bar{\boldsymbol{Z}}^T \boldsymbol{W}^T \hat{\boldsymbol{w}}_d = \lambda \mathbf{1} - N\lambda \boldsymbol{y} \tag{48}$$

Where we've used optimal scaling to cancel m_d, replacing $\boldsymbol{w}_d$ with its unit vector version $\hat{\boldsymbol{w}}_d$. This will be a useful form in future.

B.2.1 Deriving Locally Competitive Architecture

Here we derive a standard ISTA-style result from this general condition. Specifically, we can develop an equation for $\bar{\boldsymbol{z}}_d$. Starting from:

$$\frac{1}{Nm_d}\bar{\boldsymbol{X}}^T\boldsymbol{w}_d = \frac{\lambda}{N}\mathbf{1} - \lambda\boldsymbol{y} - \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}}\boldsymbol{y} \tag{49}$$

We can pull out the $\bar{\boldsymbol{z}}_d$, which always equals $\min \boldsymbol{z}_d$ when $\boldsymbol{y}$ is nonzero:

$$\bar{\boldsymbol{z}}_d = \frac{1}{m_d}\left(\frac{1}{m_d}\bar{\boldsymbol{X}}^T\boldsymbol{w}_d - \lambda\mathbf{1} + \lambda N\boldsymbol{y} + N\sum_{d'\neq d}(\bar{\boldsymbol{z}}_{d'}^T\boldsymbol{y})\bar{\boldsymbol{z}}_{d'}\right) \tag{50}$$

We can use the definition of $\boldsymbol{g}_d$ again to clarify the $\boldsymbol{y}^T\bar{\boldsymbol{z}}_d$ term:

$$\bar{\boldsymbol{z}}_{d'}^T\boldsymbol{y} = \bar{\boldsymbol{z}}_{d'}^T(\boldsymbol{y} - \frac{1}{N}\mathbf{1}) = -\bar{\boldsymbol{z}}_{d'}^T\boldsymbol{g}_d = \frac{-1}{Nm_d}\bar{\boldsymbol{z}}_{d'}^T(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{X}}^T\boldsymbol{w}_d = \frac{-1}{Nm_d}\boldsymbol{w}_{d'}^T\boldsymbol{w}_d \tag{51}$$

Which tells us that:

$$\bar{\boldsymbol{z}}_d = \frac{1}{m_d}\left(\frac{1}{m_d}\bar{\boldsymbol{X}}^T\boldsymbol{w}_d - \lambda\mathbf{1} + \lambda N\boldsymbol{y} - \frac{1}{m_d}\sum_{d'\neq d}\boldsymbol{w}_{d'}^T\boldsymbol{w}_d\bar{\boldsymbol{z}}_{d'}\right) \tag{52}$$

Using the fact that $\boldsymbol{y}$ is nonzero only when $\boldsymbol{z}_d = 0$, we can see this implements the ReLU operation, adding sufficient $\boldsymbol{y}$ to take $\boldsymbol{z}_d$ to 0. Combining this with adding the mean, $\frac{m_d}{N}$:

$$\boldsymbol{z}_d = \frac{1}{m_d^2}\text{ReLU}\left[\bar{\boldsymbol{X}}^T\boldsymbol{w}_d + m_d(\frac{m_d^2}{N} - \lambda)\mathbf{1} - \sum_{d'\neq d}\boldsymbol{w}_{d'}^T\boldsymbol{w}_d\bar{\boldsymbol{z}}_{d'}\right] \tag{53}$$

We now use the optimal scaling property, app. A.2 and denote with $\hat{\boldsymbol{w}}_d$ the unit length version of the dth column of $\boldsymbol{W}$ to find:

$$\boldsymbol{z}_d = \text{ReLU}\left[\frac{1}{m_d}\bar{\boldsymbol{X}}^T\hat{\boldsymbol{w}}_d + \mathbf{1}\left(|\min\bar{\boldsymbol{z}}_d| - \frac{\lambda}{m_d}\right) - \sum_{d'\neq d}\frac{m_{d'}}{m_d}\hat{\boldsymbol{w}}_{d'}^T\hat{\boldsymbol{w}}_d\bar{\boldsymbol{z}}_{d'}\right] \tag{54}$$

One can write this as a full population condition as follows:

$$\boldsymbol{Z} = \text{ReLU}\left[\underbrace{\text{diag}(\boldsymbol{m})^{-1}\hat{\boldsymbol{W}}^T\bar{\boldsymbol{X}}}_{\text{feedforward}} - \underbrace{\text{diag}(\boldsymbol{m})^{-1}\hat{\boldsymbol{W}}^T\hat{\boldsymbol{W}}\text{diag}(\boldsymbol{m})\bar{\boldsymbol{Z}}_{/d}}_{\text{recurrent}} + \underbrace{\theta\mathbf{1}^T}_{\text{bias}}\right] \tag{55}$$

where we've introduced the bias vector $\theta_d = |\min\bar{\boldsymbol{z}}_d| - \frac{\lambda}{m_d}$ and $\boldsymbol{Z}_{/d}$ is $\boldsymbol{Z}$ with the dth row set to 0.

This is essentially a rederivation of an ISTA fixed-point equation, as expected from theories of proximal optimality for L1-penalised problems (Hale et al., 2008). Precisely, when $||\boldsymbol{w}_d||_2 = 1$ this corresponds exactly to the Locally Competitive Algorithm in (Rozell et al., 2008), and other works (e.g. (Pehlevan & Chklovskii, 2014)). We can read from it three well-known architectural features of an optimal sparse decoder: an Affine+ReLU structure, lateral connections performing competitive inhibition, and weight tying.

There are two minor novel aspects of this derivation:

- $\boldsymbol{W}$ is defined entirely via the representation, eq. (20). Though we don't pursue it here, this naturally suggests using this self-consistent update as a dictionary learning algorithm.
- Our formulation recovers the weight tying in the encoder weights ($\hat{\boldsymbol{W}}^T$ in the feedforward term), and the recurrent weight structure, $\hat{\boldsymbol{W}}^T\hat{\boldsymbol{W}}$, but demonstrates how to scale the encoder weights when they have non-unit-norms.

Further, we can read from this the familiar conditions under which a vanilla SAE, eq. (13), recovers the optimal sparse setting: when the optimal weights are orthogonal. This result is somewhat dissatisfying, restricting the number of dictionary elements to d_x. There are slightly more relaxed settings in which decoupling the encoder and decoder weights, as is done in a vanilla SAE, is sufficient to recover the optimal sparse codes. For example, if the dot products between all features are nonpositive (as discussed in (Cui et al., 2026)), however this only gets you to maximally $2d_x$ features, and in general there remains a gap (O'Neill et al., 2024).

B.3 Perturbation within Span of SAE Features

Now let's begin by assuming that the perturbation lives within the span of the SAE features:

$$\Delta_d^T = \omega_d^T \bar{\boldsymbol{Z}} \qquad \omega_d \in \mathbb{R}^{d_z} \tag{56}$$

We derive our result in two equivalent ways. First, we restart from the perturbation equation, eq. (38), with this assumption. Second, we project the general conditions onto the span of the data, eq. (45).

B.3.1 Via Subgradients

The structure of the perturbation simplifies the change in loss, eq. (38):

$$\frac{1}{2}\Delta\mathcal{L} = -m_d N \min_{i \in \mathcal{A}_d} [\sum_{d'} \omega_{d,d'} \bar{z}_{d'}^{[i]}] - \omega_d^T \bar{\boldsymbol{Z}} (\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} \bar{\boldsymbol{X}}^T \boldsymbol{w}_d \tag{57}$$

$$= -m_d N \min_{i \in \mathcal{A}_d} [\sum_{d'} \omega_{d,d'} \bar{z}_{d'}^{[i]}] - \omega_d^T \boldsymbol{W}^T \boldsymbol{w}_d = \chi(\omega_d) \tag{58}$$

where, again, $\mathcal{A}_d$ is the set of datapoints for which $z_d^{[i]} = 0$.

Each term inside the maximum is linear in ω_d, so χ is a pointwise maximum of linear functions plus a linear term. It is therefore convex and positively homogeneous, with $\chi(\mathbf{0}) = 0$. As a result, the representation is stable to all such perturbations, $\chi(\omega_d) \geq 0$ for every ω_d, if and only if $\omega_d = \mathbf{0}$ is a global minimiser of χ, i.e.

$$\mathbf{0} \in \partial\chi(\mathbf{0}). \tag{59}$$

To compute this subdifferential we use Danskin's theorem: the subdifferential of a pointwise maximum of affine functions is the convex hull of the gradients of the functions that attain the maximum,

$$f(\boldsymbol{x}) = \max_i f_i(\boldsymbol{x}) \quad \Longrightarrow \quad \partial f(\boldsymbol{x}) = \operatorname{conv}\big\{\nabla f_i(\boldsymbol{x}) \,\big|\, f_i(\boldsymbol{x}) = f(\boldsymbol{x})\big\}. \tag{60}$$

At $\omega_d = \mathbf{0}$ every $i \in \mathcal{A}_d$ attains the maximum, so

$$\partial\chi(\mathbf{0}) = -m_d N \operatorname{conv}\big\{\bar{\boldsymbol{z}}^{[i]} \,\big|\, i \in \mathcal{A}_d\big\} - \boldsymbol{W}^\top \boldsymbol{w}_d. \tag{61}$$

The stability condition $\mathbf{0} \in \partial\chi(\mathbf{0})$ is therefore the inclusion

$$\frac{\boldsymbol{W}^\top \boldsymbol{w}_d}{-N m_d} \in \operatorname{conv}\big\{\bar{\boldsymbol{z}}^{[i]} \,\big|\, i \in \mathcal{A}_d\big\}. \tag{62}$$

If it holds, no perturbation within the span of the SAE features decreases the loss to first order. If it fails, there is a direction ω_d with $\Delta\mathcal{L} < 0$, and the representation is not locally optimal. (If $\boldsymbol{W}^\top \boldsymbol{w}_d/(-N m_d)$ lies on the boundary of the hull, some directions leave the loss unchanged to first order, and second-order terms decide.) A bit of massaging gets us to the stated condition, eq. (6):

$$-\cos\boldsymbol{\theta}_d \in \operatorname{conv}\left(\left\{\frac{\bar{\boldsymbol{z}}^{[i]}}{|\min \bar{\boldsymbol{z}}_{d'}|} \,\middle|\, i \in \mathcal{A}_d\right\}\right), \tag{63}$$

where $\cos(\theta_{d,d'}) = \hat{\boldsymbol{w}}_d^\top \hat{\boldsymbol{w}}_{d'}$.

B.3.2 Via Projection of General Condition

Alternatively, we left-multiply eq. (45) by $\bar{\boldsymbol{Z}}$:

$$\bar{\boldsymbol{Z}}\boldsymbol{g}_d = \frac{1}{N m_d} \boldsymbol{W}^T \boldsymbol{w}_d = \bar{\boldsymbol{Z}}(\frac{1}{N}\mathbf{1} - \boldsymbol{y}) = -\bar{\boldsymbol{Z}}\boldsymbol{y} \tag{64}$$

Since the only non-zero elements of $\boldsymbol{y}$ are those in $\mathcal{A}_d$, and $||\boldsymbol{y}||_1 = 1$, the right hand side is exactly a vector lying in the convex hull and we recover eq. (6).

B.4 Residual Constraints

We can spot the residuals hiding in eq. (48):

$$\bar{\boldsymbol{X}}^T \hat{\boldsymbol{w}}_d - \bar{\boldsymbol{Z}}^T \boldsymbol{W}^T \hat{\boldsymbol{w}}_d = \boldsymbol{E}^T \hat{\boldsymbol{w}}_d = \lambda\mathbf{1} - N\lambda\boldsymbol{y} \tag{65}$$

Where we've defined the residual matrix $\boldsymbol{E} = \bar{\boldsymbol{X}} - \boldsymbol{W}\bar{\boldsymbol{Z}}$. Let's take the dot-product of this with some mean-zero N-dimensional vector, $\delta\boldsymbol{z}$:

$$\hat{\boldsymbol{w}}_d^T \boldsymbol{E}\delta\boldsymbol{z} = -N\lambda\delta\boldsymbol{z}^T\boldsymbol{y} \tag{66}$$

And, recalling the properties of $\boldsymbol{y}$, we recover eq. (8). Easy!

B.5 Second Order Condition - Resurrecting Neurons

The general first-order condition, eq. (45) is necessary for a solution, $\boldsymbol{Z}$, to be a local minima, however it is not sufficient. There could be second order perturbations that decrease the loss. The clearest example of this is dead neurons waking up. They contribute a second order term to both the activity, weight, and reconstruction losses which is missed by our initial analysis. Here we derive a second order optimality condition from dead neurons waking up. Specifically, consider a representation whose dth neuron is inactive, then:

$$\bar{\boldsymbol{Z}} \to \bar{\boldsymbol{Z}} + \boldsymbol{e}_d \bar{\boldsymbol{z}}_d^T \tag{67}$$

We can calculate the change in the loss, eq. (4), to second order in $\bar{\boldsymbol{z}}_d$:

$$N^2(\min \bar{\boldsymbol{z}}_d)^2 - \bar{\boldsymbol{z}}_d^T(\lambda \mathbb{1} + \bar{\boldsymbol{Q}})^{-1}\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}(\lambda \mathbb{1} + \bar{\boldsymbol{Q}})^{-1}\bar{\boldsymbol{z}}_d \tag{68}$$

So we get a stability condition:

$$\max_{\bar{\boldsymbol{z}}_d : \mathbf{1}^T \bar{\boldsymbol{z}}_d = 0} ||\bar{\boldsymbol{X}}(\lambda \mathbb{1} + \bar{\boldsymbol{Q}})^{-1}\bar{\boldsymbol{z}}_d||_2 \leq N|\min \bar{\boldsymbol{z}}_d| \tag{69}$$

This object is invariant to scaling up or down $\bar{\boldsymbol{z}}_d$, so we can rescale to just consider cases where $\min \bar{\boldsymbol{z}}_d \geq -1$. This makes it a maximisation over a polyhedron, the feasible set: $\{\bar{\boldsymbol{z}}_d : \mathbf{1}^T \bar{\boldsymbol{z}}_d = 0, \min \boldsymbol{z}_d \geq -1\}$. The feasible set is a bounded polytope, since no element can be larger than $N-1$. The maxima of optimising a convex function over a polytope is achieved at one of the vertices, so we can just check for the vertex with highest value. The vertices are given by the settings in which all elements equal -1, except one, which equals $N-1$. Hence our condition becomes:

$$\frac{1}{N}\max_i ||\bar{\boldsymbol{X}}(\lambda \mathbb{1} + \bar{\boldsymbol{Q}})^{-1}(N\boldsymbol{e}_i - \mathbf{1})||_2 = \max_i ||\bar{\boldsymbol{X}}(\lambda \mathbb{1} + \bar{\boldsymbol{Q}})^{-1}\boldsymbol{e}_i||_2 \leq 1 \tag{70}$$

Which is nice and simple. An even cleaner result pops out if we use the Woodbury Matrix Inversion Lemma to finagle $\bar{\boldsymbol{Q}}$:

$$\bar{\boldsymbol{X}}(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1} = \frac{1}{\lambda}\bar{\boldsymbol{X}} - \frac{1}{\lambda}\bar{\boldsymbol{X}}\bar{\boldsymbol{Z}}^T(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}\bar{\boldsymbol{Z}}^T)^{-1}\bar{\boldsymbol{Z}} = \frac{1}{\lambda}(\bar{\boldsymbol{X}} - \boldsymbol{W}\bar{\boldsymbol{Z}}) \tag{71}$$

Which tells us that it all comes down to residuals!

$$\max_i ||\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{W}\bar{\boldsymbol{z}}^{[i]}||_2 \leq \lambda \tag{72}$$

C SPECIFIC DATA DISTRIBUTIONS

Here we ask when an encoding in which each neuron responds to a single source is stable, i.e. satisfies eq. (6). We first give a general sufficient condition for separable data. We then derive an exact condition for mutually exclusive categories, and ask when every choice of dictionary directions is stable.

We assume each neuron is sometimes off, $\min_i z_j^{[i]} = 0$ (else shift it down to improve the code). The demeaned activity then has $\min_i \bar{z}_j^{[i]} = -\mathbb{E}_i[z_j]$, and the normalised activity in eq. (6) is:

$$\frac{\bar{z}_j^{[i]}}{|\min \bar{z}_j|} = \frac{z_j^{[i]}}{\mathbb{E}_i[z_j]} - 1. \tag{73}$$

This equals -1 wherever neuron j is off and averages to zero over the data. Since the normalised activity is at most $\max_i z_j/\mathbb{E}_i[z_j] - 1$, a necessary condition for eq. (6) is

$$1 - \cos\theta_{k,j} \leq \frac{\max_i z_j^{[i]}}{\mathbb{E}_i[z_j]} \qquad \forall\, j \neq k. \tag{74}$$

Dictionary versus embedding geometry. The cosines in eq. (6), $\cos\theta_{k,j} = \hat{\boldsymbol{w}}_k^T \hat{\boldsymbol{w}}_j$, are between the learnt dictionary elements, not between the directions along which sources are embedded in the data. The two are closely related. Suppose the data are generated as $\boldsymbol{x}^{[i]} = \sum_k \boldsymbol{u}_k s_k^{[i]}$, and stack the $\boldsymbol{u}_k$ as the columns of $\boldsymbol{U}$ and the sources as the rows of $\boldsymbol{S}$. A single-source code, $z_k^{[i]} = a_k s_k^{[i]}$ with $a_k > 0$, has optimal dictionary:

$$\boldsymbol{W}^* = \bar{\boldsymbol{X}}(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} \bar{\boldsymbol{Z}}^T = \boldsymbol{U}\,(\lambda \mathbb{1} + \boldsymbol{\Sigma}\boldsymbol{A}^2)^{-1} \boldsymbol{\Sigma}\boldsymbol{A}, \tag{75}$$

where $\boldsymbol{A} = \mathrm{diag}(a_k)$ and $\boldsymbol{\Sigma} = \bar{\boldsymbol{S}}\bar{\boldsymbol{S}}^T$ is the (unnormalised) source covariance. If $\boldsymbol{\Sigma}$ is invertible, then as $\lambda \to 0$, $\boldsymbol{W}^* \to \boldsymbol{U}\boldsymbol{A}^{-1}$, so each $\hat{\boldsymbol{w}}_k \to \hat{\boldsymbol{u}}_k$: for small λ the dictionary geometry is the embedding geometry. At finite λ, correlated sources mix the dictionary elements slightly. For example, mutually exclusive sources have negative covariance, so orthogonally embedded categories give a nearly, but not exactly, orthogonal dictionary. We therefore state all results in terms of the dictionary cosines. Whenever a condition holds strictly for an orthogonal dictionary, it also holds, by continuity, for orthogonally embedded sources at small enough λ.

C.1 SEPARABLE DATA

The classic sparse identifiability setting models data as a sparse combination of sources, $\boldsymbol{x}^{[i]} = \sum_{k=1}^{K} \boldsymbol{u}_k s_k^{[i]}$, and asks when a dictionary in which each element encodes a single source, $z_k^{[i]} \propto s_k^{[i]}$, is optimal. One classic, if limiting, assumption from the matrix factorisation literature is separability (Donoho & Stodden, 2003): for each source j there is a datapoint i_j at which only source j is active.

Suppose the data are separable, and there is also a datapoint at which no source is active. We will show that the single-source encoding satisfies eq. (6) whenever, for every k,

$$\sum_{j \neq k} \big(1 - \cos\theta_{k,j}\big)\, \frac{\mathbb{E}_i[z_j]}{z_j^{[i_j]}} \;\leq\; 1. \tag{76}$$

To see this, consider a particular feature k. The datapoints in which each other feature is active alone, and the datapoint in which all features are off, all have $z_k = 0$, so they appear in the convex hull for neuron k. By equation 73, the all-off datapoint normalises to $-\mathbf{1}$, and i_j normalises to $-\mathbf{1} + \big(z_j^{[i_j]}/\mathbb{E}_i[z_j]\big)\boldsymbol{e}_j$. Take the weights

$$\lambda_j = \big(1 - \cos\theta_{k,j}\big)\frac{\mathbb{E}_i[z_j]}{z_j^{[i_j]}} \quad (j \neq k), \qquad \lambda_0 = 1 - \sum_{j \neq k} \lambda_j. \tag{77}$$

These are nonnegative, since $\cos\theta_{k,j} \leq 1$ and assuming equation 76, and they sum to one. The corresponding convex combination is

$$-\mathbf{1} + \sum_{j \neq k} \lambda_j \frac{z_j^{[i_j]}}{\mathbb{E}_i[z_j]}\, \boldsymbol{e}_j = -\mathbf{1} + \sum_{j \neq k} \big(1 - \cos\theta_{k,j}\big)\boldsymbol{e}_j = -\cos\boldsymbol{\theta}_k. \tag{78}$$

For an orthogonal dictionary, $\cos\theta_{k,j} = \delta_{kj}$, and equation 76 reads $\sum_{j\neq k} \mathbb{E}_i[z_j]/z_j^{[i_j]} \leq 1$. Each source must be small on average compared with its value when it is the only active source. If this holds strictly, orthogonally embedded sources are stable at small λ. Further datapoints, such as those in which several sources are active, only enlarge the convex hull. So equation 76 is sufficient but not in general necessary.

C.2 Mutually exclusive categories

Consider a categorical variable that takes one of K values, value d with probability q_d, or is inactive with probability $q_0 = 1 - \sum_d q_d$. We encode it as a K-dimensional one-hot or zero vector $\boldsymbol{x}^{[i]}$, and give each category its own neuron, $z_d^{[i]} \propto x_d^{[i]}$. Then $\mathbb{E}_i[z_d]/z_d = q_d$ whenever category d is active. We will show that if $q_0 > 0$, the one-hot encoding satisfies eq. (6) if and only if, for every k,

$$\sum_{j\neq k} q_j\big(1 - \cos\theta_{k,j}\big) \;\leq\; 1, \tag{79}$$

while if $q_0 = 0$, it is unstable whenever the dictionary elements are orthogonal or positively aligned.

To see this, again consider a feature k. The datapoints with $z_k = 0$ are the inactive ones, which normalise to $-\mathbf{1}$, and those in which some other category $j \neq k$ is active, which normalise to $-\mathbf{1} + \boldsymbol{e}_j/q_j$. A convex combination with weight λ_j on category j and λ_0 on the inactive points is $-\mathbf{1} + \sum_{j\neq k}(\lambda_j/q_j)\boldsymbol{e}_j$. This equals $-\cos\boldsymbol{\theta}_k$ if and only if $\lambda_j = q_j(1-\cos\theta_{k,j})$. These weights are nonnegative, and a nonnegative λ_0 completes them to sum to one exactly when equation 79 holds.

If $q_0 = 0$ there are no inactive datapoints, so the weights must satisfy $\sum_{j\neq k}\lambda_j = 1$ exactly. When every $\cos\theta_{k,j} \geq 0$, this sum is at most $\sum_{j\neq k} q_j = 1 - q_k < 1$, so the condition fails. (In fact, with $q_0 = 0$ the demeaned one-hot vectors sum to zero, so $\boldsymbol{\Sigma}$ is singular and, by eq. (75), the dictionary elements are linearly dependent and can never be orthogonal.) For an orthogonal dictionary with $q_0 > 0$, the left-hand side of equation 79 is $1 - q_0 - q_k < 1$, so the one-hot code is stable. Since this inequality is strict, the same holds for orthogonally embedded categories, such as the one-hot data itself, at small enough λ.

C.3 When is every geometry stable?

The previous results found simple conditions that ensured features with orthogonal dictionary elements were stable. Here we ask when every possible dictionary geometry is stable. Each condition above constrains the cosines of one neuron k at a time, and becomes harder to satisfy as any $\cos\theta_{k,j}$ decreases. The worst case is $\cos\theta_{k,j} = -1$ for all $j \neq k$, i.e. $\hat{\boldsymbol{w}}_j = -\hat{\boldsymbol{w}}_k$. This geometry arises from real data: embed every other source anti-aligned with the k-th, $\boldsymbol{u}_j = -\boldsymbol{u}_k$. Then $\boldsymbol{U}$ has rank one, so by eq. (75) every dictionary element is parallel or anti-parallel to $\boldsymbol{u}_k$, and for small enough λ (with $\boldsymbol{\Sigma}$ invertible) the signs match those of the embedding. So the encoding is stable for every dictionary geometry if and only if the condition holds with every other dictionary element anti-aligned with k. In the two cases, these conditions are:

- **Mutually exclusive** (exact):

$$\sum_{j\neq k} q_j \leq \tfrac{1}{2}\;\; \forall k \quad\Longleftrightarrow\quad q_0 + \min_k q_k \;\geq\; \tfrac{1}{2}. \tag{80}$$

  For K equally likely categories, $q_d = (1-q_0)/K$, this is $q_0 \geq \frac{K-2}{2(K-1)}$, which tends to $\frac{1}{2}$ as K grows. The more often the variable is off, the wider the range of stable geometries, until every geometry is stable once it is off about half the time.

- **Separable** (sufficient): $\sum_{j\neq k} \mathbb{E}_i[z_j]/z_j^{[i_j]} \leq \frac{1}{2}$ for every k.

  Writing $\mathbb{E}_i[z_j] = p_j\,\mathbb{E}_i[z_j \mid z_j > 0]$, each term is the frequency p_j of source j, scaled by its typical active value relative to its value at the separating datapoint. For sources whose separating value, $z_j^{[i_j]}$ is typical the condition becomes $\sum_{j\neq k} p_j \leq \frac{1}{2}$, matching the mutually exclusive case.

In both cases, positively aligned dictionary elements are easy to accommodate, and the binding constraint comes from elements anti-aligned with one another. For mutually exclusive categories, the other categories must *jointly* be rare.

C.4 Toy Test Experiments

We train a two-element dictionary on the three datasets shown in fig. 1A-C, and measure how mixed the resulting representation is using an amodularity index,

$$\text{Amodularity} = \frac{1}{2}\sum_{d=1}^{2} \min_{k\in 1,2} \left|\rho(z_d, s_k \mid s_{\neg k})\right|, \tag{81}$$

where $\rho(z_d, s_k \mid s_{\neg k})$ is the partial correlation between neuron d's activity and source k after regressing out the other source. The index is zero when each neuron's activity depends on only one source, and increases as neurons mix the two.

D Real SAE experiments

D.1 Setup

We use the SAE-lens releases `gpt2-small-res-jb` (24,576 latents per layer) and `gpt2-small-res-jb-feature-splitting` (768, 1,536, 3,072 and 6,144 latents, layer 8 only). All were trained on the `hook_resid_pre` activations of GPT2-small, on OpenWebText, with the same recipe and L1 coefficient (8×10^{-5}), and with unit-norm decoder rows. Unless stated otherwise, we use the 24,576-latent layer-8 SAE.

For each dataset we take 512 contexts of 128 tokens, which gives 65,024 tokens after dropping BOS positions (whose huge-norm residuals are outliers). Documents are shuffled and truncated to 4,000 characters, so each sample spans many documents. The datasets are:

- OpenWebText (two disjoint samples);
- C4; English Wikipedia; the Pile;
- German and French Wikipedia (one parquet shard each);
- code (`NeelNanda/code-10k`), and Python code (the same, restricted to `.py` files).

D.2 The pairwise test and the hull gauge

Write $u_d = z_d/\text{mean}(z_d)$. This equals $\bar{z}_d/|\min \bar{z}_d| + 1$ when $\min z_d = 0$, and is exactly 0 when the feature is off. eq. (6) projected onto a set S of other features reads

$$1 - \cos(\hat{w}_d, \hat{w}_S) \in \text{conv}\{u_S^{[i]} : z_d^{[i]} = 0\}.$$

Tokens at which d and every feature in S are off sit at the origin. Provided such a token exists (which is always the case in our data), the condition is therefore equivalent to the *hull gauge*

$$g(d, S) = \min\left\{ \textstyle\sum_i y_i \;:\; y \geq 0,\; \sum_{i: z_d^{[i]}=0} y_i\, u_S^{[i]} = 1 - \cos(\hat{w}_d, \hat{w}_S) \right\} \leq 1,$$

which is a linear program. $g = \infty$ means the target lies outside the cone of the points, e.g. a feature in S never fires while d is off: a strict hierarchy. For a single feature, $g = (1 - \cos\theta_{dd'})/\max_{A_d} u_{d'}$, which is the pairwise test. Because it is a maximum, a single token with the child active and the parent off can be enough to pass. A pair can also fail at the lower end of the interval, when the two features are never off together (fig. 1B); this never occurs in our data.

We test all ordered pairs of features active on more than 0.5% of tokens (at least ∼325 activations). Lowering the cutoff to 0.2% (8,800 features, 77M pairs) also gives no violations on OWT.

D.3 LAYER SHIFT

We run GPT2-small on the same OWT tokens and apply the layer-8 SAE, unchanged, to the residual stream at layers 4, 6, 7, 9 and 10. Its fraction of variance unexplained is 0.77, 0.47, 0.30, 0.25 and 1.38 respectively, against 0.14 at layer 8. The number of violating pairs, all strict hierarchies, is 146, 22, 0, 0 and 3. Every violation replicates in both halves of the tokens. Because the tokens are identical, each violating pair can be compared directly with the same pair at layer 8:

- **Layer 6.** 18 of the 22 violations share one parent (16021). At layer 8 it fires on 6–38% of tokens in the first 40 positions of a context, and almost never later; at layer 6 it fires on 100% of them. Its children are features that also fire only early in the context, and at layer 8 they fire mostly while the parent is off (e.g. 951 of 1,369 activations).
- **Layer 4.** The mirror image: parent 4760 fires on 9–22% of late tokens at layer 8, and on 99–100% of tokens beyond position 40 at layer 4, with late-position children nested inside it.
- **Layer 10.** Parent 5402 goes from 1–3% of tokens at layer 8 to 84–99% at every position.

The tokens on which the parents are off are ordinary tokens ("the", commas, "of"), with as many other features active as elsewhere, so the effect is not an artefact of inactive tokens.

D.4 JOINT VIOLATIONS

The joint search is not exhaustive. As targets d (the feature whose inactive tokens form the hull), we take only the 25 features whose largest pairwise gauge is highest, i.e. those whose pair constraints are already closest to failing. For each, we test the joint condition against the k features that individually come closest to violating ($k = 1, 3, 10, 30, 100$), excluding features that already violate pairwise. For every violating set, we greedily remove features while the set still violates, which gives a minimal violating set. Results:

- OWT, C4, English Wikipedia and the Pile: no set violates; the median gauge at $k = 100$ is 0.19–0.26.
- German and French: no set violates, although gauges are the highest of any dataset (median at $k = 100$: 0.60 and 0.57).
- Code: 2 of the 25 targets violate, with minimal sets of 20–25 features; the worst pair within each set has gauge ≤ 0.14.
- Python code: 3 of 25 violate, with minimal sets of 21–80 features.

The joint gauge exceeds the pairwise ones because the members' activity outside the parent comes from largely disjoint tokens, and the weights must sum to at most 1. When there is no overlap, the joint gauge is the sum of the pairwise gauges; for one code set the pairwise gauges sum to 1.8, while the joint gauge is 1.09. In the code sets, the target fires on essentially every indentation space, and the members are features that each fire at a different depth within the run of indentation spaces.

D.5 WHAT THE VIOLATING FEATURES DO ON OWT

For each dataset violation, we look up the same features on 130k OWT tokens (both samples):

- **German and French.** The parents respond to foreign words inside English text ("Le Monde Diplomatique", "résumé"). The children are generic English grammar features ("that / which / who"; "this / that"; "as / like"; sentence-final newlines; commas), and each fires almost always while the parent is off (e.g. 2,115 of 2,129 activations, and 626 of 626).
- **Python.** The parent fires on stray space tokens in prose. The child fires on islands and beaches ("Sapelo Island"), never with the parent on.

The hierarchies are therefore not present in the SAE's training data. They arise because, off-distribution, one feature saturates and others fire only within it.

D.6 Feature-residual condition

Let E be the demeaned residuals, u a principal direction of E with variance σ^2, and $s = E^\top u$. Eq. (7) with $\delta z = s$ gives, for any residual,

$$-\sigma^2 \cos(\hat{w}_d, u) \in \lambda\,[\min_{A_d} s,\ \max_{A_d} s],$$

the full-rank version of eq. (8). For each feature d and each of the top 50 residual directions, we compute the smallest λ at which the pair passes:

$$\lambda_{\min}(d, u) = \sigma^2 |\cos| \,/\, \big(\max_{A_d} s \text{ if } \cos < 0,\ \text{else } |\min_{A_d} s|\big),$$

and set $\lambda^* = \max \lambda_{\min}$. The condition holds for every $\lambda \geq \lambda^*$, so the SAE's residuals are consistent with its own training strength exactly when the residual break ratio $\lambda^*/\lambda_{\text{train}}$ is at most 1.

Training λ. The SAEs were trained with the per-element MSE, normalised by the centred input norm, plus $c\sum_d |z_d|$ with $c = 8 \times 10^{-5}$. In the convention of eq. (1), this is $\lambda_i = c\, d_{\text{in}}\, \|x_i - \bar{x}\|/2$, with median 2.32 on OWT. As an independent check, the median of $\hat{w}_d^\top e$ over each feature's active tokens, which equals λ at an optimum (eq. (48)), is 2.38 (5–95% range across features: 2.07–2.70).

Results.

- OWT, C4, English, German and French Wikipedia: 0.19–0.35, at every width. The largest residual directions on German and French are larger than on OWT (standard deviation 4–9 vs. $\sim$3), but none is aligned with a feature in a violating way.
- The Pile: 0.45–0.81.
- Code: 1.0–1.9. The violation is borderline for the main SAE (4 features, exceeding the threshold by 1–5%) and clear for the 768-latent SAE (18 features, by up to $1.9\times$). All violating features are aligned with residual PC1, which runs from indentation spaces to identifier fragments. On code, 60–100% of their activations are on whitespace. On OWT most respond to formatting and byte-level tokens (GPT-2's byte fragments of curly quotes and apostrophes, e.g. in "don't"; separators such as "====", and stray spaces).
- Layer shift (main SAE): 0.78, 0.70, 0.63 and 0.52 at layers 4, 6, 7 and 9, so no violations. At layer 10 the value is 17, but the SAE no longer reconstructs the input there (FVU 1.38; residual standard deviation 66 vs. $\sim$3 at layer 8).

E Optimal Antipodal Dense Encodings

It is hard to solve the dictionary learning problem analytically. Here, we consider a toy case in which we *can* describe everything, which we use to understand how an SAE encodes dense variables, section 6. We consider an SAE trained on a scalar input. We find that with nonnegativity this leads to antipodal encoding of dense features - i.e. optimally the representation contains two neurons encoding the positive and negative component of the scalar. This turns out to be a special case of the wide-convex limiting solution considered in app. F, except, thanks to the structure of the discovered solutions, the encoder architecture constraint is automatically satisfied, so in this case it is also a statement about vanilla SAEs.

E.1 KKT Conditions

In this analysis we will begin by deriving the KKT conditions for eq. (21). We enforce the nonnegativity of $\boldsymbol{Z}$ with a matrix of Lagrange multipliers, $\boldsymbol{P}$:

$$\mathcal{M} = \mathbf{1}^T \boldsymbol{Z}^T \boldsymbol{Z} \mathbf{1} + \operatorname{Tr}[(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} \bar{\boldsymbol{X}}^T \bar{\boldsymbol{X}}] - 2 \operatorname{Tr}[\boldsymbol{P}^T \boldsymbol{Z}] \tag{82}$$

Taking the derivative with respect to $\boldsymbol{Z}$ gives:

$$\frac{1}{2}\frac{\partial \mathcal{M}}{\partial \boldsymbol{Z}} = \boldsymbol{Z}\mathbf{1}\mathbf{1}^T - \bar{\boldsymbol{Z}}(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} \bar{\boldsymbol{X}}^T \bar{\boldsymbol{X}} (\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} (\mathbb{1} - \frac{1}{N}\mathbf{1}\mathbf{1}^T) - \boldsymbol{P} \tag{83}$$

Further, it is not hard to see that $\mathbf{1}$ is an eigenvector of $(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1}$, and, hence, since $\bar{\boldsymbol{X}}$ is demeaned:

$$\bar{\boldsymbol{X}}(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} \mathbf{1} \propto \bar{\boldsymbol{X}} \mathbf{1} = 0 \tag{84}$$

Which allows us to drop the term that ends $\mathbf{1}\mathbf{1}^T$. Hence our final KKT conditions are:

$$\boldsymbol{Z}\mathbf{1}\mathbf{1}^T - \bar{\boldsymbol{Z}}(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{X}}^T\bar{\boldsymbol{X}}(\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1} - \boldsymbol{P} = \mathbf{0}, \qquad \boldsymbol{Z} \geq 0, \quad \boldsymbol{P} \geq 0, \quad \boldsymbol{Z} \odot \boldsymbol{P} = \mathbf{0} \tag{85}$$

Where $\odot$ denotes elementwise multiplication. In general these conditions are difficult to manipulate, but we shall make progress in the case of a scalar input.

E.2 Smallest Optimal Scalar Encoding Uses (at most) 2 Neurons

First, we shall show that there are, at most, two unique neural responses, $\boldsymbol{z}_d$, in the population.

If each datapoint is a scalar, $x^{[i]}$, we can spot a single vector is repeated in the KKT conditions: $\boldsymbol{a} = (\lambda\mathbb{1} + \bar{\boldsymbol{Z}}^T\bar{\boldsymbol{Z}})^{-1}\bar{\boldsymbol{x}}$. Using this vector, the KKT conditions become:

$$\boldsymbol{Z}\mathbf{1}\mathbf{1}^{\mathbf{T}} - \bar{\boldsymbol{Z}}\boldsymbol{a}\boldsymbol{a}^T = \boldsymbol{P} \tag{86}$$

Breaking this down by neuron:

$$\boldsymbol{z}_d^T\mathbf{1}\mathbf{1} - \bar{\boldsymbol{z}}_d^T\boldsymbol{a}\boldsymbol{a} = \mu_d\mathbf{1} - \alpha_d\boldsymbol{a} = \boldsymbol{p}_d \tag{87}$$

This is great, because each neuron contributes only through two scalars, $\mu_d = \boldsymbol{z}_d^T\mathbf{1}$ and $\alpha_d = \bar{\boldsymbol{z}}_d^T\boldsymbol{a}$. Otherwise everything is governed by the shared vector, $\boldsymbol{a}$. The sharing of this vector across the population places heavy constraints. Call the set of indices at which neuron d has zero activity $\mathcal{A}_d$. Then, for all $i \notin \mathcal{A}_d$, $p_d^{[i]} = 0$ by the KKT conditions and:

$$a^{[i]} = \frac{\mu_d}{\alpha_d} \tag{88}$$

Hence, the vector $\boldsymbol{a}$ is broken down into parts. Only in those parts whose elements equal the fixed value $\frac{\mu_d}{\alpha_d}$ can neuron d have non-zero activity. Any other neuron that has a different value of $\frac{\mu_d}{\alpha_d}$ must then be off during these datapoints.

Now consider a second neuron, index d', with its own set of inactive datapoints, $\mathcal{A}_{d'}$. Unless $\frac{\mu_{d'}}{\alpha_{d'}} = \frac{\mu_d}{\alpha_d}$, $\mathcal{A}_{d'}^c$ and $\mathcal{A}_d^c$, the complements of the two sets where the neurons are active, must be disjoint; let's consider the case where they are disjoint. Consider a datapoint $i \notin \mathcal{A}_d$, the other neuron must be off, $z_{d'}^{[i]} = 0$, therefore $p_{d'}^{[i]} \geq 0$. This tells us that:

$$p_{d'}^{[i]} = \mu_{d'} - \alpha_{d'}a^{[i]} = \mu_{d'} - \alpha_{d'}\frac{\mu_d}{\alpha_d} \geq 0 \tag{89}$$

Considering the converse points, where neuron d is off and d' is on, we derive similarly that:

$$\mu_d - \frac{\mu_{d'}\alpha_d}{\alpha_{d'}} \geq 0 \tag{90}$$

We know both μ_d and $\mu_{d'}$ are positive (since $\boldsymbol{z}_d \geq 0$ and $\mu_d = \mathbf{1}^T\boldsymbol{z}_d$). Consider the case when α_d and $\alpha_{d'}$ have the same sign, then the only way for these two inequalities to be satisfied is if $\frac{\mu_d}{\alpha_d} = \frac{\mu_{d'}}{\alpha_{d'}}$, i.e. the neurons in fact share the same part of $\boldsymbol{a}$ vector. Alternatively they can have a different sign, and the two inequalities are satisfied. Extending this argument to more neurons we see they'll always fall into these two groups.

These groups don't have to be exactly one neuron. For example, the population could contain two neurons that were scaled copies of one another, or they could split the datapoints between them, with one neuron fitting one part, another neuron the rest. However, the optimal solution can be reached using any population with at least two neurons.

E.3 Precise Representation

Now we shall derive the functional form of the two-neuron optimal representation.

The definition of $\boldsymbol{a}$ tells us that:

$$\lambda\boldsymbol{a} + \alpha_+\bar{\boldsymbol{z}}_+ + \alpha_-\bar{\boldsymbol{z}}_- = \bar{\boldsymbol{x}} \tag{91}$$

This tells us, first, that $\mathbf{1}^T \boldsymbol{a} = 0$. We can then pin down the remaining $\{a_\pm, \mu_\pm, \alpha_\pm\}$ a little more:

$$\alpha_+ = \boldsymbol{a}^T \bar{\boldsymbol{z}}_+ = \boldsymbol{a}^T \boldsymbol{z}_+ = a_+ \sum_i z_+^{[i]} \tag{92}$$

$$\mu_+ = \mathbf{1}^T \boldsymbol{z}_+ = \sum_i z_+^{[i]} \tag{93}$$

Hence:

$$\alpha_+ = a_+ \mu_+ \qquad \alpha_+^2 = \mu_+^2 \tag{94}$$

Identical arguments work for μ_- and α_-, and since we know that $\alpha_\pm$ have to have different signs, we end up with:

$$a_+ = 1 = \frac{\mu_+}{\alpha_+} \quad a_- = -1 = \frac{\mu_-}{\alpha_-} \tag{95}$$

Next, let's relate the representation back to the data. Using eq. (91):

$$z_+^{[i]} = \left[\frac{1}{\mu_+} (\bar{x}^{[i]} - \lambda - \frac{\mu_-^2 - \mu_+^2}{N}) \right]_+ \tag{96}$$

$$z_-^{[i]} = \left[\frac{1}{\mu_-} (-\bar{x}^{[i]} - \lambda + \frac{\mu_-^2 - \mu_+^2}{N}) \right]_+ \tag{97}$$

This is a pretty implicit representation of $\mu_\pm$. We use the following self-consistency relation to numerically find the 'effective threshold' θ, using $\mu_\pm = \mathbf{1}^T \boldsymbol{z}_\pm$:

$$\theta = \frac{\mu_-^2 - \mu_+^2}{N} = \mathbb{E}_i \left[[-\bar{x}^{[i]} - \lambda + \theta]_+ - [\bar{x}^{[i]} - \lambda - \theta]_+ \right] \tag{98}$$

So we create an array of θ values, and choose the value that minimises the difference between the left and right hand side. Having found a θ value, $\mu_\pm$ can be easily calculated from using:

$$\mu_\pm^2 = \mathbf{1}^T [\pm \bar{x}^{[i]} - \lambda \mp \theta]_+ \tag{99}$$

Unless λ is chosen very particularly, there will be no datapoint that lies exactly on the boundary, ensuring that strict complementary slackness applies, and hence making this solution the global optima. This completes our optimal modular representation.

E.4 When is it one neuron?

Our final question is where is the threshold between using one or two neurons? We shall answer this by proposing an optimal single neuron encoding, and finding when it is unstable to the addition of an extra neuron.

Consider the following optimal single neuron encoding, of the form identified in the previous section:

$$\boldsymbol{z} = a[\bar{\boldsymbol{x}} - \theta \mathbf{1}]_+ = a(\bar{\boldsymbol{x}} - \theta \mathbf{1} + \Lambda) \tag{100}$$

Where Λ is a vector that implements the ReLU, i.e.:

$$\Lambda^{[i]} = \begin{cases} 0 & \text{if} \quad \bar{x}^{[i]} - \theta > 0 \\ -(\bar{x}^{[i]} - \theta) & \text{else} \end{cases} \tag{101}$$

The mean-zero version is then:

$$\bar{\boldsymbol{z}} = a(\bar{\boldsymbol{x}} + \Lambda - \frac{1}{N} \mathbf{1}^T \Lambda \mathbf{1}) \tag{102}$$

Now we can plug this into the first order optimality condition, eq. (48), to get:

$$\bar{\boldsymbol{x}} + N\lambda \boldsymbol{y} - \lambda \mathbf{1} = \bar{\boldsymbol{z}} w = wa(\bar{\boldsymbol{x}} + \Lambda - \frac{1}{N} \mathbf{1}^T \Lambda \mathbf{1}) \tag{103}$$

Hence, simply by equating factors, we find:

$$wa = 1, \qquad \Lambda = N\lambda \boldsymbol{y} \tag{104}$$

Summing the indices on the second equation tells us:

$$\frac{1}{N}\sum_i [-(\bar{x}^{[i]} - \theta)]_+ = \lambda \tag{105}$$

Let's assume λ is smaller than the distance between points, then this really constrains how far $\bar{x}^{[i]}$ can travel below the threshold!

Let's make a simple assumption that it is only below threshold when $\bar{x}^{[i]} = \min \bar{\boldsymbol{x}}$. Then this is telling us that:

$$\theta = \frac{\lambda}{q} - |\min \bar{\boldsymbol{x}}| \tag{106}$$

where q is the proportion of points at which $x^{[i]} = \min \bar{\boldsymbol{x}}$.

Finally, let's measure when this is stable to the addition of a new neuron. Using eq. (72) we see that a condition is that:

$$|\min \bar{\boldsymbol{x}} - wa(\theta - \lambda)| = \lambda \frac{1-q}{q} \leq \lambda \tag{107}$$

Hence $q > \frac{1}{2}$, the variable is at least half-sparse.

For larger λ this condition will change slightly, in ways that could be calculated using a more elaborate version of this analysis.

E.5 Empirical Verification

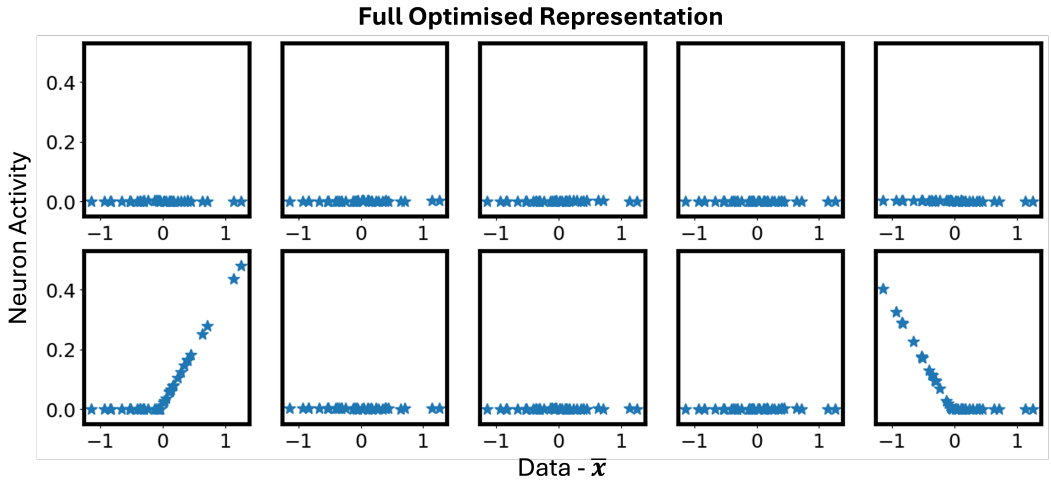


Figure 6: We optimise ten nonnegative neurons to encode a scalar variable by minimising eq. (4). Only two neurons have non-zero activity and their functional form matches the prediction, fig. 4. Somehow optimisation selects this solution from amongst the degenerate set of optimal solutions.

We empirically test these claims by optimising a nonnegative encoding to minimise eq. (4). We input a scalar and run gradient descent on $\boldsymbol{Z}$ subject to nonnegativity. Figure 4 shows the comparison of the two non-zero neurons to the predictions from app. E.3. We include the full 10 neuron representation we optimised to illustrate how cleanly the representation uses only two neurons, fig. 6. The choice of using two neurons, rather than another of the degenerately optimal solutions, somehow results from optimisation.

F WIDE-DICTIONARY SETTING

In this section we consider the structure of the optima when there is no constraint on the number of neurons. In this case the problem is convex, app. A.4. Bach et al. (2008) study a related problem and find that one-neuron-per-datapoint is a global minima . However, this problem has a slight non-triviality: we saw in app. E that no matter how many neurons we added, the optimal solution could be achieved using at most 2 neurons. Here we show that indeed, one-neuron-per-datapoint is a global minima, but further, that the narrowest solution that achieves this global minima is governed by the number of rays from the optimal bias $\boldsymbol{b}$ required to hit all the datapoints further than λ from $\boldsymbol{b}$.

Consider a representation that activates at most a single feature per datapoint:

$$z_d^{[i]} = \begin{cases} a_d \geq 0 & \text{if} \quad d = i \\ 0 & \text{else} \end{cases} \tag{108}$$

Let's consider the original version of the problem, eq. (12). Then the optimal choice of $\boldsymbol{w}_d$ is a unit vector aligned with $\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{b}$, making the loss proportional to:

$$\sum_i ||\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{b}||^2 \left(1 - \frac{a^{[i]}}{||\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{b}||}\right)^2 + 2\lambda \sum_i a^{[i]} \tag{109}$$

Following the arguments in the main paper, section 7, there must be at most one active neuron per datapoint in the globally minimal solution whose dictionary vector is aligned along $\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{b}$ for the $\bar{\boldsymbol{x}}^{[i]}$ on which it is active. This constrains the active sets of each neuron to those along rays from $\boldsymbol{b}$ and tells us that a global minima is exactly the set of ray solutions.

Optimising with respect to $a^{[i]}$ and taking into account the nonnegativity we get:

$$a^{[i]} = \max(||\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{b}|| - \lambda, 0) \tag{110}$$

This tells us exactly that datapoints within λ of $\boldsymbol{b}$ are encoded with the zero vector, hence not contributing to the number of neurons needed to reach the minimal width global minima.

We find that $\boldsymbol{b}$ can be found as the solution to the following optimisation problem:

$$\boldsymbol{b} = \arg\min_b \sum_i \ell_\lambda(||\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{b}||) \quad \text{where} \quad \ell_\lambda(||\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{b}||) = \begin{cases} ||\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{b}||^2 & ||\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{b}|| \leq \lambda \\ 2\lambda||\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{b}|| - \lambda^2 & ||\bar{\boldsymbol{x}}^{[i]} - \boldsymbol{b}|| > \lambda \end{cases} \tag{111}$$

Which is a Huber-like interpolation between a squared and linear penalty. In the limit of small λ this becomes the geometric median eq. (9).

1D Dataset We derived the representation of 1D data using a convoluted KKT argument, app. E, however it can be much more simply seen here. We can shuffle activity amongst neurons pointing in the same direction without changing the loss. For 1D data there are at most two directions. Shuffling activity so that only two neurons are active, or one if possible, we recover the same solution.