How Optimality Structures Sparse Dictionaries
A Theory for Understanding SAE Representations

William Dorrell [1]

Abstract

Sparse Autoencoders (SAEs) have found success parsing neural representations into interpretable concepts, providing a basis for understanding and control. However, what exactly SAEs extract, and, correspondingly, the scientific conclusions we can draw from them, are not obvious. Empirically, the proof is in the pudding: SAEs learn interpretable features. Theoretically, we lack a clear account of what properties a 'concept' must satisfy for an SAE to extract it. There has been extensive identifiability work studying the conditions under which sparse coding recovers ground-truth features; however, these approaches tends to focus on simple data-generating models (e.g. sparse independent features) which poorly approximate the internet-swallowing language-model representations on which SAEs are trained. Here, avoiding data-generating models, we ask simply what properties any dictionary learning optimum must satisfy. Concretely, we extend local optimality analyses (Gribonval & Schnass, 2010) to the nonnegative joint-optimisation problem that vanilla SAEs approximate, and derive constraints relating optimal SAE features to their distributions. We use these constraints to explain a range of observed SAE behaviours - hierarchical splitting & absorption, the structure of residuals, and dense antipodal features - each reflecting how L1+nonnegativity interact with data to structure optimal dictionaries. Finally, we construct a novel large-dictionary convex problem and explore the wide atom-per-datapoint limit. In sum, we hope to tease model assumptions from unexpected observations, letting us learn more from SAEs' successes and provide principles for designing their successors.

[1]Kempner Institute, Harvard University. Correspondence to: William Dorrell <dorrellwec@gmail.com>.

Mechanistic Interpretability Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea, 2026. *Copyright 2026 by the author.*

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 of which contributes 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, a field which tries to understand neural networks as a systems neuroscientist does the brain, has driven a surge of interest in dictionary learning by using it as a tool to understand neural networks. This is motivated by two observations. First, neural network representations often seem to encode human-interpretable concepts as linear directions (Mikolov et al., 2013; Arora et al., 2016). Second, these concepts are often sparse, and there are more of them than the representation has neurons, so the network is forced to encode them in 'superposition' (Arora et al., 2018; Elhage et al., 2022). This ensures that single neurons are 'polysemantic' - they encode multiple concepts - hindering approaches that interpret each neuron's behaviour. Rather, conceptualising representations as sparse linear combinations of concepts - the 'linear representation hypothesis' - naturally motivates using dictionary learning to disentangle the underlying concepts. Indeed, different flavours of Sparse Autoencoders (SAEs), a dictionary learning variant, have proven effective at extracting meaningful concepts from the internal representations of Large Language Models (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 and use them to 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 a phenomenon called 'feature splitting': a concept identified by a smaller SAE, such as 'base64', splits into multiple concepts 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 more natural. Should we then interpret feature splitting as evidence that neural networks do not encode hierarchies?

Thankfully, the answer is likely 'no'. Rather, feature splitting represents an unintended consequence of applying dictionary learning to hierarchically structured data. This example illustrates the broader point motivating our work (among others - see Section 2). Dictionary learning is a tool that assumes representations can be well-modelled as sparse linear combinations of components. When we examine the results of applying dictionary learning techniques, like SAEs, to neural networks, we are not just seeing symptoms of the object of interest - neural network behaviour - but also the assumptions we 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 precisely link our observations to underlying network behaviour, and provide targets for developing tools that more precisely reflect our goals.

In this work, we take steps towards such an understanding. Our main theoretical result is a set of conditions that a representation must obey to be an optima of a dictionary learning problem, analogous to earlier results 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 show how these constraints explain or predict a series of SAE behaviours - hierarchical splitting & absorption, the structure of residuals, or dense antipodal features. Additionally, we highlight a reformulation of the dictionary learning problem that, in the many-neuron limit, is convex, which shows that SAEs will keep refining their categorisations until each ray of data gets its own dictionary atom (Bach et al., 2008). In sum, we provide theoretical 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.

2. Related Work

Throughout we draw extensively on empirical SAE work. Here, we discuss three other related strands.

Dictionary Learning Identifiability There has been extensive work on the identifiability of dictionary learning, drawing on related results in compressed sensing (sparse coding with a fixed dictionary) (Candes & Tao, 2006; Donoho, 2006; Donoho & Elad, 2003), or 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, 2015; 2018; Gribonval et al., 2015; Cohen & Gillis, 2019; Wang et al., 2020). Here, motivated by the heterogeneity of real 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 dictionary learning problem. For a dictionary learning problem similar to that approxiamted by SAEs, Gribonval & Schnass (2010) derive local optimality conditions without data assumptions. We draw on these techniques and apply them to the most SAE-relevant version of dictionary learning. Further, we use the results of Bach et al. (2008) to reformulate our dictionary learning problem and study the large width limit.

Sparse Autoencoders Theory We highlight three approaches that try, like us, to theoretically understanding SAE representations. First, Hindupur et al. (2025) study how architectural constraints determine the concepts SAEs can extract, a restriction we largely ignore. Second, two recent works study the identifiability of SAEs: conditions under which they recover a set of true sparse features. Cui et al. (2026) show that, in the limit of extreme sparsity, features whose embedding dot-products are non-positive can be recovered. Tang et al. (2025) instead 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 works are analogues of older dictionary learning work applied to an SAE architecture. Here, in contrast, we study the underlying dictionary learning problem that SAEs approximate; we therefore eschew architectural statements in favour of general optimality conditions to understand SAE behaviour. Third, follow-up work on feature splitting, and its cousin feature absorption, linked these phenomenon to the interplay of sparsity and hierarchy (Till, 2024; Chanin et al., 2024). We build on these works, making the same link more robustly, permitting generalisations.

Neural Coding There have been decades of neuroscientific work that, like us, relate neural responses to solutions of optimisation problems (Laughlin, 1981; Atick & Redlich, 1990; Simoncelli & Olshausen, 2001), including, famously, dictionary learning (Olshausen & Field, 1996). We draw technical inspiration from a particular strand of this work: Sengupta et al. (2018); Dorrell et al. (2026) demonstrate the convexity of interesting non-negative neural coding problems by lifting to the space of representational similarity

matrices. This is a particular example of the Semi-Definite relaxation, familiar for example from polynomial optimisation (Shor, 1987; Lasserre, 2009; Anstreicher, 2012; Pena et al., 2015) or other work on matrix factorisation (Bach et al., 2008), which we will also use to demonstrate convexity.

3. Problem Formalisation, 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, 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:

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

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 Equation (1).

Encoder Architecture While the linearity of the decoder, from $\boldsymbol{z}^{[i]}$ to $\boldsymbol{x}^{[i]}$, reflects the key representational hypothesis, the encoder that extracts $\boldsymbol{z}^{[i]}$ from $\boldsymbol{x}^{[i]}$ 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 places additional constraints on the learnt SAE concepts. Motivated partly by this gap between ideal dictionary learning representations (e.g. solutions to Equation (1)) and those learnt by a standard SAE, various works considered alternate encoders (Hindupur et al., 2025), or losses (Fel et al., 2025), including classic sparse coding algorithms (Costa et al., 2025; O'Neill et al., 2024), or nonlinearities such as JumpReLU or TopK (Gao et al., 2024; Bussmann et al., 2024; Rajamanoharan et al., 2024). These choices certainly modify optimal SAE representations.

For simplicity and generality, we focus on the vanilla dictionary learning problem, Equation (1). While some of our results, e.g. the following reformulations, hold if we add the encoder constraint, $\boldsymbol{z}^{[i]} = \text{ReLU}(\boldsymbol{W}_{\text{enc}}\boldsymbol{x}^{[i]} + \boldsymbol{b}_{\text{enc}})$, in general there will be differences. Despite this, we find our conclusions match empirical patterns from various SAEs. Theoretically, this can be partially understood via the limiting settings in which SAEs exactly solve Equation (1). See Section A.1 for further discussion.

Loss Reformulation In Section A.2, we use a scaling symmetry argument from Bach et al. (2008) to develop two equivalent formulations of the original problem, Equation (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:

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

Hence, while changing from Equation (1) to Equation (2) aids optimisation (Templeton et al., 2024), it does not meaningfully 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:

$$\mathcal{L}_3 = \mathbb{E}_i\left[||\boldsymbol{W}\boldsymbol{z}^{[i]} + \boldsymbol{b} - \boldsymbol{x}^{[i]}||_2^2\right] + \frac{\lambda}{N}\sum_{d=1}^{d_z}(||\boldsymbol{w}_d||_2^2 + ||\boldsymbol{z}_d||_1^2), \quad \min_{\boldsymbol{W},\{\boldsymbol{z}^{[i]}\}_{i=1}^N,\boldsymbol{b}} \mathcal{L}_3 \quad \text{subject to} \quad \boldsymbol{z}^{[i]} \geq 0 \tag{3}$$

Now the loss is an unconstrained quadratic in the dictionary matrix, $\boldsymbol{W}$, allowing us to optimise it out, Section 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, ridge regression tells us that 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 reformula-

tion:

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

Convex Reformulation In Section A.4, we note that the problem, Equation (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 it's demeaned version $\bar{\boldsymbol{Q}}$:

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

$\boldsymbol{Z} \geq 0$ 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), meaning that if $d_z \geq N$, thereby relaxing the rank constraint, this optimisation problem is convex, ensuring local minima are global. In English, in the unrealistic setting when there are more SAE neurons than datapoints, the dictionary learning problem can be reformulated as convex. Relative to the convex problems explored by Bach et al. (2008), nonnegativity frames a particularly clean convex problem using $\mathcal{CP}^N$, though 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 $\boldsymbol{z}^{[i]} = \operatorname{ReLU}(\boldsymbol{W}_{\text{enc}} \boldsymbol{x}^{[i]} + \boldsymbol{b}_{\text{enc}})$, 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 (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 SAE 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 limit of dictionary learning, and find that almost all data structures seem to lead to extreme feature splitting, reinforcing the vital dependence of discovered concepts on SAE width.

4. Feature Constraints, Separability, and Hierarchical Splitting/Absorption

To begin, we describe how local optimality forces features to inactivate separately, allowing us to understand feature splitting and absorption.

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 Section 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 Section B.2, which recovers the classic ISTA-like algorithm (Daubechies et al., 2010) 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 Section 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.

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

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 features' directions and joint-distribution.

Intuition Consider an SAE with 2 neurons, meaning there's only one cosine similarity to consider: $\cos(\theta_{12}) = \hat{\boldsymbol{w}}_1^T \hat{\boldsymbol{w}}_2$. The red regions in Figure 1A illustrate the relevant convex hulls for two independent sparse variables. In this case the convex hulls span the entire $[-1, 1]$ region, so every choice of $\cos(\theta_{12})$ is allowed under Equation (6), reflecting the fact that sparse, independent features are stably encoded by SAEs. However, it also motivates the types of struc-

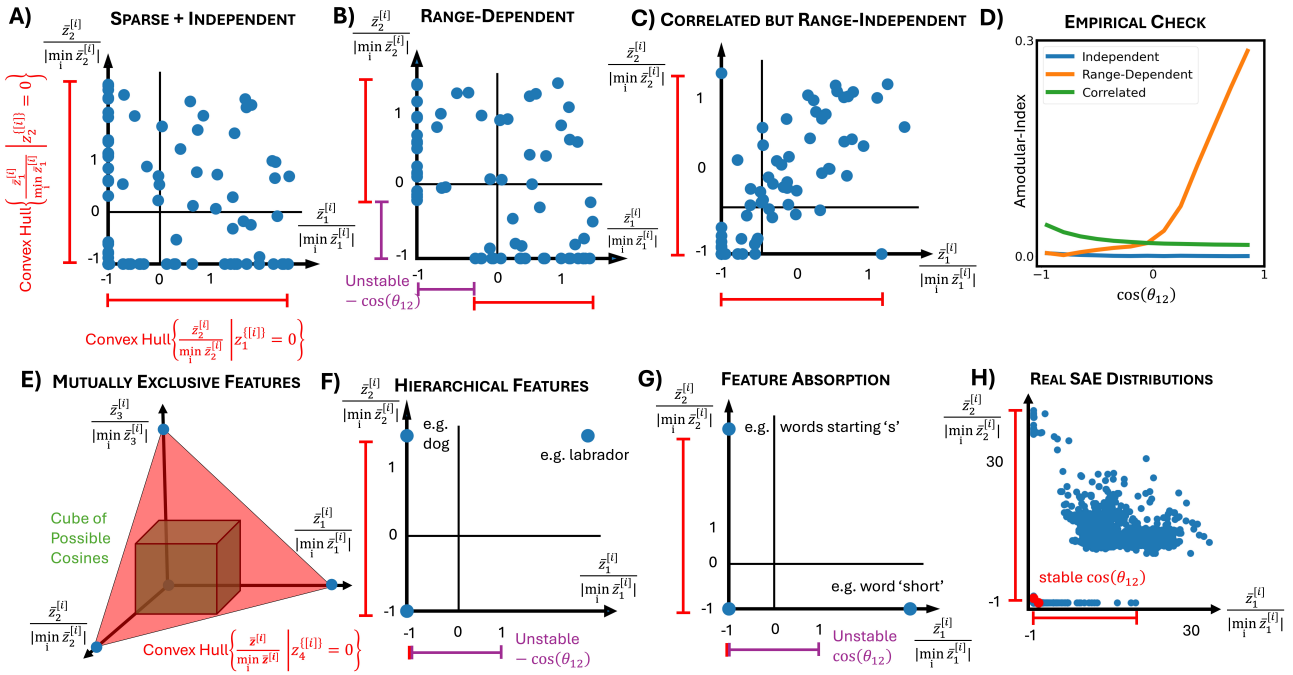

Figure 1. **A)** We plot two putatively optimal sparse independent features, normalised according to Equation (6). The relevant convex hulls (red regions) are the region explored by one feature while the other is inactive. In this case the convex hulls cover $[-1, 1]$, so these features are stable for 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 (0 means each dimension is perfectly encoded in one dictionary element). We see meaningful amodularity only in the cases predicted from the convex hulls, Section C.3. **E)** Mutually exclusive features are stable for all cosine vectors. **F)** Hierarchical features are unstable under all cosine vectors except perfect alignment, whereas **(G)** feature absoprtion turns the hierarchy into mutually exclusive features, stabilising them.**H)** We test these relations on pairs of real SAE features and find no contradictions amongst 7.1 millions pairs.

ture that will destabilise SAEs. Consider two features that never inactivate together; the convex hull will lose its minimal region, destabilising the features whose directions are highly aligned[1], Figure 1B. Alternatively, two features can be very highly correlated while still satisfying Equation (6), as long as they activate and inactivate together, Figure 1C. Figure 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.

Optimal Features An SAE solution should never have a neuron with $|\min \bar{\boldsymbol{z}}_d| \geq \max \bar{\boldsymbol{z}}_d$ (if it does, reorient to $-\bar{\boldsymbol{z}}_d$ to reduce the loss!). With this, we present three implications of Equation (6):

- A sufficient condition for a set of features to satisfy Equation (6) is that, while each feature is off, all others explore the extremities of their range, as in Figure 1A and C. This is independent of feature directions, illustrating the centrality of range properties.
- Mutually exclusive categories of any dimensionality satisfy Equation (6), Figure 1E, Section C.1, matching the SAE propensity for categorical structure.
- Working from the other direction, if the features are roughly orthogonal, the condition tells us that, to be optimal, the relevant convex hulls has to engulf the zero vector. This can be achieved, for example, by a dataset in which all features activate significantly in isolation, the familiar separability condition from the Nonnegative Matrix Factorisation literature (Donoho & Stodden, 2003), Section C.2. More generally, most existing identifiability results tackle this near-orthogonal regime (e.g. (Gribonval & Schnass, 2010; Wu & Yu, 2015; Wang et al., 2020)), so their results could perhaps be mapped onto this geometric picture.

Hierarchical Splitting & Absorption Two of the most striking SAE observations are feature splitting, and its cousin, feature absorption. In feature splitting, a broad concept that appears in a narrow SAE, like 'maths', is seen to split into more fine-grained concepts in larger SAEs, like 'geometry'

[1]Intuitively, since they are explaining similar data they can be merged, and since they don't inactivate together, keeping the combination nonnegative won't incur too much activity cost.

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 active 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 leads 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, Equation (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 Figure 1F. Since all labradors are dogs, the distribution of 'labrador' while 'dog' is inactive, one of the convex hulls appearing in Equation (6), is a single point corresponding to both features being inactive. This ensures that, 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 Figure 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 been again been turned into a set of mutually exclusive categories, Figure 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 use SAE-lens to perform a test of the predicted relationships for an SAE with $\sim$ 25000 features trained on block 8 of the residual stream of GPT2-small (Bloom, 2024; Bloom et al., 2024) and evaluated on the same OpenWebText dataset use to train the SAE (Interestingly, we find counterexamples when evaluating on the pile-10k dataset, reflecting features that our theory predicts would not exist had the SAE been trained on pile-10k). We test all features that activate on at least 0.5% of the dataset ($\sim$ 4000 feature) and verify that the cosine-inclusion relation holds for each of the 7.1 million resulting feature-pairs (clean example Figure 1H, details: Section D). A fuller test would require assessing relations beyond pairs, which is feasible using a linear program, but we don't pursue it here.

5. Error Residuals

Next, we consider first-order conditions on the feature-residual relationships that preclude the dictionary from learning concepts that are too 'similar' to the residuals, in a certain sense.

Constraining Feature-Residual Relationships In Section 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. Define the residual matrix, $\boldsymbol{E} = \bar{\boldsymbol{X}} - \boldsymbol{W}\bar{\boldsymbol{Z}}$, then:

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

where $\delta\boldsymbol{z}$ is an arbitrary mean-zero response vector. This shows a natural dependence on λ: harsher regularisation means looser constraints on the residuals.

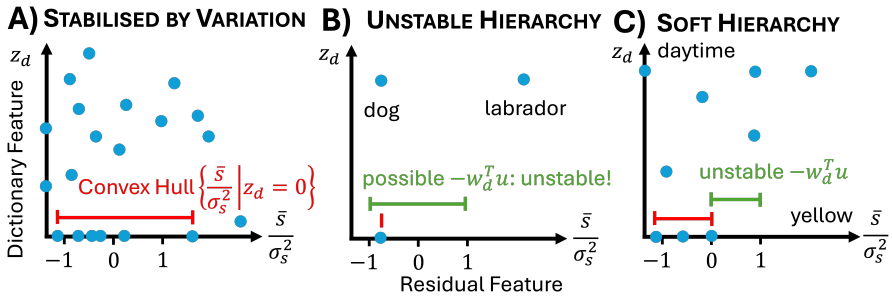

Figure 2. **A)** We plot a dictionary against missing feature; if the missing feature varies widely enough while the dictionary is inactive, it can be stably left in the residuals Equation (7). **B)** Hierarchical features don't survive this check, while **(C)** soft hierarchies directionally fail.

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, quantified by the left and right hand-side respectively. $\hat{\boldsymbol{w}}_d^T \boldsymbol{E}$ 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 the change in the reconstruction. The right-hand side quantifies how much L1 cost will be needed to construct such an encoding. Putting these together, a direction, $\delta \boldsymbol{z}$, that positively aligned with the residuals along $\boldsymbol{w}_d$ and was active whenever the dth neuron was inactive would destabilise the solution.

Let's say the residual data is rank one, $\boldsymbol{E} = \boldsymbol{u}\bar{\boldsymbol{s}}^T$ ($||\boldsymbol{u}|| = 1$ without loss of generality). This means we are leaving one concept out in the residual, when is that allowed? Orthogonal to $\bar{\boldsymbol{s}}$, Equation (7) is trivially satisfied, so choosing $\epsilon = \bar{\boldsymbol{s}}$:

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

where we've defined $\sigma_s^2 = \frac{1}{N}||\boldsymbol{s}||^2$. This is telling us that, 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. 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, Figure 2A. Similarly, mutually exclusive features satisfy this criterion.
- Range-dependency destabilise 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, Figure 2B.
- Softer hierarchies are destabilised in one direction. For example, perhaps a lower level feature ('yellow') only activates maximally with a higher level feature ('daytime'), but activates weakly at other times. A 'daytime' neuron would be destabilised by the 'yellow' residual if their encoding directions align ($\hat{\boldsymbol{w}}_d^T \boldsymbol{u} > 0$) but not if they antialign, Figure 2C.

6. Dense Antipodal Feature Pairs

Dense features provide a simple case study in SAEs behaving strangely when applied to datasets whose assumptions they disagree with. 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 cluster into antipodal pairs (i.e. 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, Figure 3A?

We find this is not the case. In Section E we study a simple example of a vanilla SAE encoding a single dimension, i.e. the input data, $x^{[i]}$, can be mapped to a scalar. 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),

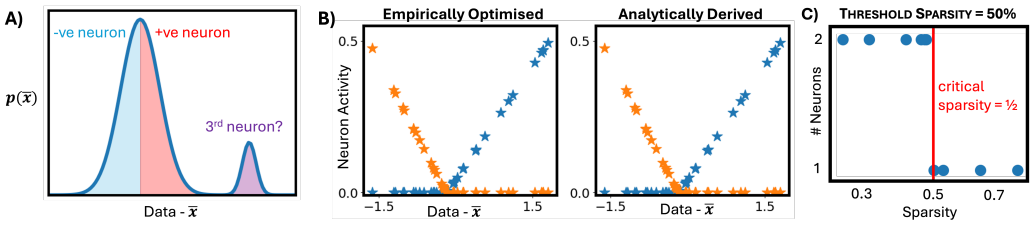

Figure 3. **A)** While splitting a dense variable into positive and negative halves seems intuitive, perhaps outlying clumps should have their own neuron? **(B)** We find this is not the case, an optimal solution to Equation (4) for rank 1 data exists with at most 2 non-zero neurons. This theoretically predicted low-neuron representation, Section E, fits exactly. For our simulations we optimised a dictionary with 10 elements/neurons, but the representation only at most used 2, Figure 5. **C)** Theory predicts that a variable that is more than half sparse should be encoded with only one active neuron: empirics agree.

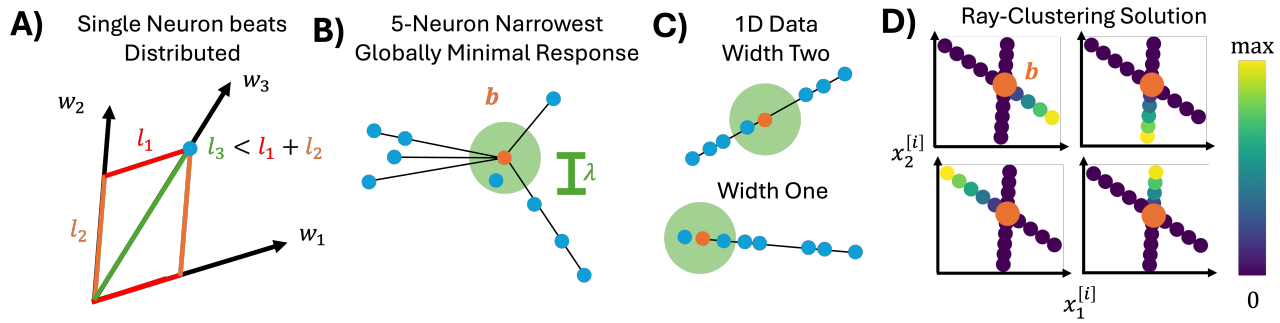

Figure 4. **A)** If a datapoint activates two neurons, and neurons are in abundance, it can be equally well-explained by a third that is aligned and hence uses less activity. **B)** If datapoints share rays from the optimal bias, $\boldsymbol{b}$, it can be explained using only one neuron, narrowing 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 Figure 3. **(D)** We train a 30 neuron dictionary on this 2D cross dataset. The geometric median lands in the middle of the cross, and the optimal dictionary uses only four active neurons to explain the four rays of the data, as predicted.

Figure 3B. 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, Figure 3C. 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 on a broader phenomenon: ray-clustering.

7. Ray-Clustering and the Wide Convex Limit

Feature splitting raises a question: as you increase SAE size, will the splitting continue *ad infinitum*? In short, yes. In the wide convex limit there exists a globally optimal solution with a dictionary atom for every datapoint (Bach et al., 2008). For a simple route to this conclusion, consider the original objective, Equation (1). Imagine a solution in which two distinct dictionary atoms (i.e. with different dictionary vectors $\boldsymbol{w}_d$) activate for a single datapoint, Figure 4A. 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, Figure 4A. Hence, at a global minima each datapoint must activate at most one latent.

Narrow Global Minima However, the solution discussed in Section 6 teaches us that globally minimal solutions can sometimes be reached using much narrower dictionaries. What characterises the narrowest 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 4B. In Section F, we derive the optimal bias, $\boldsymbol{b}$, implicitly as the solution to an optimisation problem. In the perfect reconstruction limit ($\lambda \to 0$) it is given by the geometric median of the dataset, defined as:

$$\boldsymbol{b} = \arg\min_{\boldsymbol{b}} \sum_d ||\bar{\boldsymbol{x}}_d - \boldsymbol{b}||_2 \tag{9}$$

This object generalises the standard notion of median to multidimensional data and has been studied for centuries. Under imperfect reconstruction $\boldsymbol{b}$ is a slight generalisation of this quantity, Section F.

Perfect Reconstruction Hence, as $\lambda \to 0$, the narrowest globally minimal solution is determined by the number of rays required to hit every datapoint from $\boldsymbol{b}$, the geometric median. This explains Section 6: 1-dimensional data requires either one or two rays depending on whether $\boldsymbol{b}$ lies within or at the extreme of the data distribution, Figure 4C. In 1D $\boldsymbol{b}$ is simply the median, so if more than 50% of the data lies on an extreme point it is the median and only one ray is required, else, two are.

Imperfect Reconstruction At finite λ multiple distinct datapoints may simply be approximated as $\boldsymbol{b}$ without any features activating. In Section B.5, we develop a second-order condition that determines whether a solution is stable to adding a new neuron:

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

i.e. a solution is stable to new neurons if the residuals for all datapoints are below λ. Interestingly, if a solution is a minima for the problem with d_z neurons and satisfies this condition, we can pad it with as many dead neurons as we need to reach N, and it becomes a global minima of the convex problem Equation (5). Hence, Equation (10) determines whether a local minima of the dictionary learning problem is the wide global minima. From this condition we can see that datapoints that are within a λ-ball of $\boldsymbol{b}$ won't receive their own dictionary atom. Hence, the width of the narrowest

global optima is given by the number of rays required to hit all datapoints further than λ from $\boldsymbol{b}$, Figure 4B-D, where $\boldsymbol{b}$ is defined implicitly as the solution to an optimisation problem APP, similar to Equation (9).

This is mostly a mathematical curiosity of dictionary learning, indeed, the encoder constraint means the wide vanilla SAE representation will differ. However, it provides clean characterisation of the global optima towards which dictionary learning is pushing, which might motivate principled alternative algorithms that behave better.

8. Discussion

In this paper we considered how optimality in a dictionary learning problem, Equation (1), necessarily structures learnt dictionaries, and we have used this to understand a slew of sparse autoencoder phenomenology, including hierarchical splitting & absorption, the structure of residuals, dense antipodal features, and the neuron-per-datapoint-limit. This illustrates a broader point: observations are filtered through the tools we use. Understanding how these tools behave is therefore key, and we hope the understanding provided will be useful both for interpretation, and in principled development of SAE successors.

One shortcoming is the limited role we have given encoder architecture. Clearly, architecture shapes learnt solutions (Hindupur et al., 2025), and the gap between SAEs and the optimal sparse codes we consider is real (Barin Pacela et al., 2026). However, as a relatively general problem with implications beyond mechanistic interpretability (e.g. neural coding), we think dictionary learning is a natural starting point. Future work could usefully apply similar tools to characterise how architectural choices, like ReLU vs. JumpReLU, shape solutions. More generally, deriving architecture from optimisation, as in Section B.2.1, is an influential strand of work (Amari, 1977; Hopfield, 1982; Oja, 1982; Linsker, 1988; Bell & Sejnowski, 1995; Olshausen & Field, 1996; Rozell et al., 2008; Pehlevan & Chklovskii, 2019). It would be interesting to derive architectures for optimisation problems inspired by the heterogeneity measured in data, such as hierarchies (Bricken et al., 2023) or multidimensional latents (Engels et al., 2024).

Finally, motivated partly by the changing nature of features as SAE width is increased and surprising properties like feature absorption there has been concern about the future of SAEs as a tool for mechanistic interpretability (Leask et al., 2025). Narrowly, their existing uses, such as for controlled steering (Durmus et al., 2024) and unsupervised concept discovery, motivate a better theoretical understanding of their behaviour. More broadly, the extent to which paradigmatic ideas like the linear representation hypothesis are useful remains unclear. Since SAEs form the natural technical counterpart to this hypothesis, better understanding them can only help in laying out the ways this hypothesis succeeds and fails.

Reproducibility All mathematical arguments are included in the appendix. Code: `https://github.com/WilburDoz/SAE_Identifiability`.

Acknowledgements I'd like to thank a large crowd of people for talking with me about this paper, especially for their guidance in exploring areas that were new to me: Priscila de Azevedo Drummond, James Whittington, Demba Ba, Shubham Choudhary, Sumedh Hindupur, Kyle Hsu, Luke Hollingsworth, Hadas Orgad, Bahareh Tolooshams, Valérie Costa. Further, special thanks go to Rodrigo Carrasco-Davis and Andy Arditi for comments on a draft of this paper, to Jin Hwa Lee for discussions of relevant SAE literature, and especially to Basile Confavereux for being the perfect partner-in-crime, were it not for all the very important other things he has to do ;)

References

Amari, S.-I. Neural theory of association and concept-formation. *Biological cybernetics*, 26(3):175–185, 1977.

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

Arad, D., Mueller, A., and Belinkov, Y. 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.

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

Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495, 2018.

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

Bach, F., Mairal, J., and Ponce, J. Convex sparse matrix factorizations. *arXiv preprint arXiv:0812.1869*, 2008.

Barin Pacela, V., Joshi, S., Camacho, I., Lacoste-Julien, S., and Klindt, D. Stop probing, start coding: Why linear probes and sparse autoencoders fail at compositional generalisation. *arXiv e-prints*, pp. arXiv–2603, 2026.

Barrett, D. G., Deneve, S., and Machens, C. K. Optimal compensation for neuron loss. *Elife*, 5:e12454, 2016.

Bell, A. J. and Sejnowski, T. J. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

Berman, A. and Shaked-Monderer, N. *Completely positive matrices*. World Scientific, 2003.

Bloom, J. 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.

Bloom, J., Tigges, C., Duong, A., and Chanin, D. Saelens. https://github.com/decoderesearch/SAELens, 2024.

Boyd, S. and Vandenberghe, L. Convex optimization. *Cambridge UP*, 2004.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2, 2023.

Bussmann, B., Leask, P., and Nanda, N. Batchtopk sparse autoencoders. *arXiv preprint arXiv:2412.06410*, 2024.

Bussmann, B., Nabeshima, N., Karvonen, A., and Nanda, N. Learning multi-level features with matryoshka sparse autoencoders. *arXiv preprint arXiv:2503.17547*, 2025.

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

Chanin, D., Wilken-Smith, J., Dulka, T., Bhatnagar, H., Golechha, S., and Bloom, J. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. *arXiv preprint arXiv:2409.14507*, 2024.

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

Costa, V., Fel, T., Lubana, E. S., Tolooshams, B., and Ba, D. From flat to hierarchical: Extracting sparse representations with matching pursuit. *arXiv preprint arXiv:2506.03093*, 2025.

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

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Daubechies, I., DeVore, R., Fornasier, M., and Güntürk, C. S. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 63(1):1–38, 2010.

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

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

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

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

Dorrell, W., Latham, P. E., and Whittington, J. Convex efficient coding. *arXiv preprint arXiv:2601.10482*, 2026.

Dumitrescu, B. and Irofti, P. *Dictionary learning algorithms and applications*. Springer, 2018.

Durmus, E., Tamkin, A., Clark, J., Wei, J., Marcus, J., Batson, J., Handa, K., Lovitt, L., Tong, M., McCain, M., et al. Evaluating feature steering: A case study in mitigating social biases. *Anthropic Research*, 2024.

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

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Engels, J., Michaud, E. J., Liao, I., Gurnee, W., and Tegmark, M. Not all language model features are one-dimensionally linear. *arXiv preprint arXiv:2405.14860*, 2024.

Fel, T., Lubana, E. S., Prince, J. S., Kowal, M., Boutin, V., Papadimitriou, I., Wang, B., Wattenberg, M., Ba, D., and Konkle, T. Archetypal sae: Adaptive and stable dictionary learning for concept extraction in large vision models. *arXiv preprint arXiv:2502.12892*, 2025.

Gao, L., la Tour, T. D., Tillman, H., Goh, G., Troll, R., Radford, A., Sutskever, I., Leike, J., and Wu, J. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

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

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

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

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

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

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

Hindupur, S. S. R., Lubana, E. S., Fel, T., and Ba, D. Projecting assumptions: The duality between sparse autoencoders and concept geometry. *arXiv preprint arXiv:2503.01822*, 2025.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

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

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

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

Lasserre, J. B. *Moments, positive polynomials and their applications*, volume 1. World Scientific, 2009.

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

Leask, P., Bussmann, B., Pearce, M., Bloom, J., Tigges, C., Moubayed, N. A., Sharkey, L., and Nanda, N. Sparse autoencoders do not find canonical units of analysis. *arXiv preprint arXiv:2502.04878*, 2025.

Linsker, R. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

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

Mikolov, T., Yih, W.-t., and Zweig, G. 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.

Motzkin, T. S. Beiträge zur theorie der linearen ungleichungen. *(No Title)*, 1936.

Oja, E. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

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

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

O'Neill, C., Gumran, A., and Klindt, D. Compute optimal inference and provable amortisation gap in sparse autoencoders. *arXiv preprint arXiv:2411.13117*, 2024.

Pehlevan, C. and Chklovskii, D. B. 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.

Pehlevan, C. and Chklovskii, D. B. Neuroscience-inspired online unsupervised learning algorithms: Artificial neural networks. *IEEE Signal Processing Magazine*, 36(6):88–96, 2019.

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

Rajamanoharan, S., Lieberum, T., Sonnerat, N., Conmy, A., Varma, V., Kramár, J., and Nanda, N. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024.

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

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

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

Sengupta, A., Pehlevan, C., Tepper, M., Genkin, A., and Chklovskii, D. Manifold-tiling localized receptive fields are optimal in similarity-preserving neural networks. *Advances in neural information processing systems*, 31, 2018.

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

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

Sun, X., Stolfo, A., Engels, J., Wu, B., Rajamanoharan, S., Sachan, M., and Tegmark, M. Dense sae latents are features, not bugs. *arXiv preprint arXiv:2506.15679*, 2025.

Tang, Y., Saini, H., Yao, Z., Liao, Y., Li, Q., Du, M., and Liu, D. On the theoretical foundation of sparse dictionary learning in mechanistic interpretability. *arXiv preprint arXiv:2512.05534*, 2025.

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

Team, G., Riviere, M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N. L., McDougall, C., MacDiarmid, M., Freeman, C. D., Sumers, T. R., Rees, E., Batson, J., Jermyn, A., Carter, S., Olah, C., and Henighan, T. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

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

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

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

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

Wu, S. and Yu, B. Local identifiability of l_1-minimization dictionary learning: a sufficient and almost necessary condition. *arXiv preprint arXiv:1505.04363*, 2015.

Wu, S. and Yu, B. 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 N datapoints, $\boldsymbol{x}^{[i]} \in \mathbb{R}^{d_x}$, we seek each 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 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, Equation (11), and sparsity, we use the following optimisation problem, Equation (1) in the main text:

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

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 Equation (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, Equation (12) and Equation (13), represents the 'vanilla sparse autoencoder' (SAE) problem. The Affine+ReLU encoder structure is important for our context: it can constrain the learnt SAE representations. However, we will ignore it for three reasons:

1. In some settings solutions with and without the Affine+ReLU encoder constraint, Equation (13), are equivalent (for example, when dictionary elements are orthogonal Section B.2), ensuring our results on dictionary learning apply exactly to vanilla SAEs, though note that this seems not to be true for SAEs trained on large-scale network representations.

2. We have introduced the vanilla SAE, but this is only one of a zoo of SAE architectures. Various changes to the architecture (Bussmann et al., 2024; Rajamanoharan et al., 2024; Hindupur et al., 2025) and loss (Fel et al., 2025; Costa et al., 2025; Bussmann et al., 2025) have been proposed, and it is unlikely that there will be a canonical version. That said, they all share the core principles represented by dictionary learning, so Equation (12) seems a good 'model organism' to study. Further, dictionary learning is a widely used algorithm, e.g. in models of neural processing (Olshausen & Field, 1996), whereas SAEs are not, suggesting insights into dictionary learning are more likely to generalise.

3. The dictionary learning problem is mathematically simpler.

Future work could usefully consider the impact of different SAE architectures using the techniques studied here. Here, we proceed with our analysis of dictionary learning

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, Equation (3) in the main text, that initially appears quite different to the original dictionary learning problem Equation (12):

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

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:

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

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 Equation (14) we recover Equation (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 eventual settle to the optimal scaling σ_d^*, solving the original problem, Equation (14), is equivalent to solving Equation (17) then rescaling optimally. Finally, however, Equation (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, Equation (12). One thing this illustrates is that the SAE improvements reported by Templeton et al. (2024) upon moving from Equation (12) to Equation (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 Equation (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 $\boldsymbol{Z}$:

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

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

A.4. Convexity in the $d_z \geq N$ Limit

Here we show that we can rewrite Equation (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 more neurons than datapoints, $d_z \geq N$. 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:

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

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

Positivity Constraint The set of dot product matrices with unconstrained ranks formed from vectors that are themselves positive ($\boldsymbol{Q} = \boldsymbol{Z}^T \boldsymbol{Z}$ for $\boldsymbol{Z} \geq 0$) form a convex set called the set of completely positive matrices (Berman & Shaked-Monderer, 2003). However, when the number of neurons is smaller than the number of datapoints (the standard setting), $\boldsymbol{Q}$ must additionally be constrained to have rank lower than d_z. This is a non-convex constraint, though relaxing to the closest convex envelope, the nuclear norm, would be an interesting direction for future work. More generally, it is interesting to precisely describe how nonconvexity arises in this problem.

Weight and Reconstruction Loss The weight and reconstruction loss can be clearly 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:

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

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

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

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:

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

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

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 Section 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 pertubation:

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

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, Equation (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.

If we write 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:

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

This condition can be equivalently expanded as:

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

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

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

Where $\mathbb{1}_{\mathcal{A}_d} \in \mathcal{R}^{|\mathcal{A}_d|\times T}$ 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 $\boldsymbol{g}_d$:

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

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, Equation (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 $\boldsymbol{g}_d$:

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

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:

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

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

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

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.

Uniqueness of Optima Satisfying this condition is necessary for being a local optima, when is it sufficient? If the dual certificate $\boldsymbol{y} > 0$, i.e. is strictly positive, the minima is sharp to first order, and so this becomes a sufficient condition.

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 $\boldsymbol{z}_d$ isn't, 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, $\frac{\mu_d}{N}$:

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

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

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

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 $\theta_d = |\min \boldsymbol{z}_d| - \frac{\lambda}{\boldsymbol{m}}$ 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, Equation (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, Equation (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, Equation (38), with this assumption. Second, we project the general conditions onto the span of the data, Equation (45).

B.3.1. Via Subgradients

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

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

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

where, again, $\mathcal{A}_d$ is the set of datapoints for which $z_d^{[i]} = 0$. One way to proceed now is using the subderivative. The subderivative of the minima of a set of convex functions is within the convex hull of the derivatives of the functions that achieve that minima:

$$f(x) = \min_i f_i(x) \qquad \partial f(x) \in \text{Convex Hull}(\{\partial f_i(x) | f_i(x) = f(x)\}) \tag{59}$$

Hence:

$$\frac{1}{2}\frac{\partial\Delta\mathcal{L}}{\partial\omega_d} = -m_d N\boldsymbol{f}_d - \boldsymbol{W}^T\boldsymbol{w}_d \tag{60}$$

where $\boldsymbol{f}_d$ is the subderivative. a vector that lives in the span of the datapoints that minimise neuron d's activity:

$$\frac{\partial \min_{i\in\mathcal{A}_d}[\sum_{d'}\omega_{d,d'}\bar{z}_{d'}^{[i]}]}{\partial\omega_d} = \boldsymbol{f}_d \in \text{Convex Hull}(\{\bar{\boldsymbol{z}}^{[i]} | i \in \mathcal{A}_d\}) \tag{61}$$

So for stability we must simply ask whether the following inclusion is satisfied:

$$\frac{\boldsymbol{W}^T\boldsymbol{w}_d}{-Nm_d} \in \boldsymbol{f}_d \tag{62}$$

If so, the change to the loss is nonnegative, else the representation is not locally optimal. A bit of massaging gets us to the stated condition Equation (6):

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

where $\cos(\theta_{d,d'}) = \hat{\boldsymbol{w}}_d^T\hat{\boldsymbol{w}}_{d'}$.

B.3.2. Via Projection of General Condition

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

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

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, hence we recover Equation (63).

B.4. Residual Constraints

We can spot the residuals hiding in Equation (48):

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

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, ϵ:

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

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

B.5. Second Order Condition - Resurrecting Neurons

The general first-order condition, Equation (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, Equation (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 $\min \boldsymbol{z}_d \geq -1$. 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 discuss some specific data distributions and examine the conditions under which they satisfy the feature-feature constraint Equation (6).

C.1. Mutually Exclusive Categories

Consider a mutually exclusive categorical variable. The variable takes one of K different values, or it is inactive. Denote with q_d the probability of the dth category, and with q_0 the probability of being inactive. We will encode this in a K-dimensional vector, $\boldsymbol{x}^{[i]}$ that is either a one-hot encoding of the category, or the zero vector if the category is inactive. We will ask whether an encoding in which each category is given its own neuron, $z_d^{[i]} \propto x_d^{[i]}$, is stable.

The key question for stability is whether the cube of feasible dot products is engulfed by the relevant convex hull. Consider one of the convex hulls identified in Equation (6), for example, the datapoints in which $x_K^{[i]} = 0$. To examine this convex hull we need to perform the correct normalisation. Denoting the set of indices on which category d is active with $\mathcal{I}_d$:

$$\frac{\bar{x}_d^{[i]}}{|\min \bar{\boldsymbol{x}}_d|} = \begin{cases} \frac{1-q_d}{q_d} & i \in \mathcal{I}_d \\ -1 & \text{else} \end{cases} \tag{73}$$

Further, recalling that, for all stable features $|\min \boldsymbol{z}_d| < \max \boldsymbol{z}_d$, else multiply by -1 to reduce the loss and satisfy this. This constrains $q_d < \frac{1}{2}$, which we shall assume is true. Hence the relevant convex hull is:

$$\text{Convex Hull}\left(\left\{ \frac{\bar{\boldsymbol{z}}^{[i]}}{|\min \bar{\boldsymbol{z}}^{[i]}|} \middle| z_K^{[i]} = 0 \right\}\right) = \text{Convex Hull}\left(\left\{ \boldsymbol{0}, \frac{1-q_d}{q_d}\boldsymbol{e}_d : d \neq K \right\}\right) \tag{74}$$

We're going to try and reach the all ones vector, the most difficult to reach point in the space:

$$\sum_{d=1}^{K-1} \boldsymbol{e}_d \frac{1-q_d}{q_d}\lambda_d = \sum_{d=1}^{K-1} \boldsymbol{e}_d \frac{1-q_d}{q_d}\frac{q_d}{1-q_d}\left(\sum_{d=1}^{K-1}\frac{1-q_d}{q_d}\right) = \left(\sum_{d=1}^{K-1}\frac{1-q_d}{q_d}\right)\sum_{d=1}^{K-1}\boldsymbol{e}_d \tag{75}$$

Since $q_d < \frac{1}{2}$, $\frac{1-q_d}{q_d} > 1$, so the sum is guaranteed to be pointing along the all ones direction, but beyond the all ones vector. By appropriately mixing in the origin point we can recover any point on this ray, including the all ones vertex. We can play equivalent games to reach all the other vertices of the cube using the relevant subset of features. For example, to reach the $(1, 1, 0, 0, ...)$ vertex we can combine just $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ weighted by $\frac{q_1}{1-q_1}$ etc. Having reached every vertex, we can combine arbitrarily to reach every point on the interior.

C.2. Separable Data

The classic sparse identifiability condition studies dataset that have been created from some sparse linear combination of factors:

$$\boldsymbol{x}^{[i]} = \sum_{k=1}^{K} \boldsymbol{u}_k s_k^{[i]} \tag{76}$$

We then ask: when is a dictionary in which each dictionary element encodes a single $s_k^{[i]}$ optimal? One classic, if very limiting, data assumption from the matrix factorisation literature is separability (Donoho & Stodden, 2003). This means that there exists at least one datapoint where only one of the sources, $s_k^{[i]}$, is non-zero, for each source k.

Let's take the very simple case: let's assume that in this single active datapoint $s_k^{[i]} = |\min \boldsymbol{s}_k|$. Further, that each $\boldsymbol{u}_k$ is orthogonal there is a datapoint at which no source is active. Under Equation (6), the dictionary encoding in which each recover a source is stable if the origin is a member of the relevant convex hull. Doing a equally weighted convex combination of all active features gets a nonnegative vector pointing along the all ones direction, adding in an appropriate weighting of the datapoint in which no feature is active gets us to the origin. More elaborate versions of this could be constructed that apply to less contrived settings.

C.3. Experiments

We train a two-element dictionary on the three datasets shown in Figure 1A-C. To measure amodularity we do the following. We remove the component of each datapoint along the direction of one of the dimensions of the data, and measure the

correlation with the other. We take the minima of these two numbers, and average this across the two dictionary elements. This encodes how much the features are mixing in the dictionary.

D. Real SAE Experiments

We downloaded a pre-trained SAE from block 8 of GPT2-small using sae-lens (Bloom, 2024; Bloom et al., 2024), and studied its representation on the OpenWebText dataset. Specifically, we checked whether the pairwise feature-feature condition was satisfied for all SAE features that activated a significant amount: on more than 0.5% of the tokens tested corresponding to 128 tokens. To do this we extracted the SAE representation on a portion of the dataset and explicitly tested whether the cosine similarity lay within the relevant convex hull, Figure 1. We found that in all cases the conditions were satisfied, and we show an interesting looking example in Figure 1H.

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 Section 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 Equation (21). We enforce the nonnegativity of $\boldsymbol{Z}$ with a matrix of Lagrange multipliers, $\boldsymbol{P}$:

$$\mathcal{M} = \mathbf{1}^T \boldsymbol{Z}^T \boldsymbol{Z} \mathbf{1} + \mathrm{Tr}[(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} \bar{\boldsymbol{X}}^T \bar{\boldsymbol{X}}] - 2\,\mathrm{Tr}[\boldsymbol{P}^T \boldsymbol{Z}] \tag{77}$$

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

$$\frac{1}{2}\frac{\partial \mathcal{M}}{\partial \boldsymbol{Z}} = \boldsymbol{Z}\mathbf{1}\mathbf{1}^T - \bar{\boldsymbol{Z}}(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} \bar{\boldsymbol{X}}^T \bar{\boldsymbol{X}} (\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} (\mathbb{1} - \frac{1}{T}\mathbf{1}\mathbf{1}^T) - \boldsymbol{P} \tag{78}$$

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

$$\bar{\boldsymbol{X}}(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} \mathbf{1} \propto \bar{\boldsymbol{X}} \mathbf{1} = 0 \tag{79}$$

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

$$\boldsymbol{Z}\mathbf{1}\mathbf{1}^T - \bar{\boldsymbol{Z}}(\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} \bar{\boldsymbol{X}}^T \bar{\boldsymbol{X}} (\lambda \mathbb{1} + \bar{\boldsymbol{Z}}^T \bar{\boldsymbol{Z}})^{-1} - \boldsymbol{P} = \mathbf{0}, \qquad \boldsymbol{Z} \geq 0, \quad \boldsymbol{P} \geq 0, \quad \boldsymbol{Z} \odot \boldsymbol{P} = \mathbf{0} \tag{80}$$

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:

$$\boldsymbol{Z}\mathbf{1}\mathbf{1}^{\mathbf{T}} - \bar{\boldsymbol{Z}}\boldsymbol{a}\boldsymbol{a}^T = \boldsymbol{P} \tag{81}$$

Breaking this down by neuron:

$$\boldsymbol{z}_d^T \mathbf{1}\mathbf{1} - \bar{\boldsymbol{z}}_d^T \boldsymbol{a}\boldsymbol{a} = \mu_d \mathbf{1} - \alpha_d \boldsymbol{a} = \boldsymbol{p}_d \tag{82}$$

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:

$$a^{[i]} = \frac{\mu_d}{\alpha_d} \tag{83}$$

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:

$$p_{d'}^{[i]} = \mu_{d'} - \alpha_{d'} a^{[i]} = \mu_{d'} - \alpha_{d'} \frac{\mu_d}{\alpha_d} \geq 0 \tag{84}$$

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

$$\mu_d - \frac{\mu_{d'} \alpha_d}{\alpha_{d'}} \geq 0 \tag{85}$$

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:

$$\lambda \boldsymbol{a} + \alpha_+ \bar{\boldsymbol{z}}_+ + \alpha_- \bar{\boldsymbol{z}}_- = \bar{\boldsymbol{x}} \tag{86}$$

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:

$$\alpha_+ = \boldsymbol{a}^T \bar{\boldsymbol{z}}_+ = \boldsymbol{a}^T \boldsymbol{z}_+ = a_+ \sum_i z_+^{[i]} \tag{87}$$

$$\mu_+ = \mathbf{1}^T \boldsymbol{z}_+ = \sum_i z_+^{[i]} \tag{88}$$

Hence:

$$\alpha_+ = a_+ \mu_+ \qquad \alpha_+^2 = \mu_+^2 \tag{89}$$

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

$$a_+ = 1 = \frac{\mu_+}{\alpha_+} \quad a_- = -1 = \frac{\mu_-}{\alpha_-} \tag{90}$$

Next, let's relate the representation back to the data. Using Equation (86):

$$z_+^{[i]} = \left[\frac{1}{\mu_+} (\bar{x}^{[i]} - \lambda - \frac{\mu_-^2 - \mu_+^2}{T}) \right]_+ \tag{91}$$

$$z_-^{[i]} = \left[\frac{1}{\mu_-} (-\bar{x}^{[i]} - \lambda + \frac{\mu_-^2 - \mu_+^2}{T}) \right]_+ \tag{92}$$

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

$$\theta = \frac{\mu_-^2 - \mu_+^2}{T} = \mathbb{E}_i \left[[-\bar{x}^{[i]} - \lambda + \theta]_+ - [\bar{x}^{[i]} - \lambda - \theta]_+ \right] \tag{93}$$

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:

$$\mu_\pm^2 = \mathbf{1}^T [\mp \bar{x}^{[i]} - \lambda \mp \theta]_+ \tag{94}$$

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:

$$\boldsymbol{z} = a[\bar{\boldsymbol{x}} - \theta \mathbf{1}]_+ = a(\bar{\boldsymbol{x}} - \theta \mathbf{1} + \Lambda) \tag{95}$$

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

$$\Lambda^{[i]} = \begin{cases} 0 \quad \text{if} \quad \bar{x}^{[i]} - \theta > 0 \\ -(x^{[i]} - \theta) \quad \text{else} \end{cases} \tag{96}$$

The mean-zero version is then:

$$\bar{\boldsymbol{z}} = a(\bar{\boldsymbol{x}} + \Lambda - \frac{1}{N} \mathbf{1}^T \Lambda \mathbf{1}) \tag{97}$$

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

$$\bar{\boldsymbol{x}} + N\lambda \boldsymbol{y} - \lambda \mathbf{1} = \bar{\boldsymbol{z}} w = wa(\bar{\boldsymbol{x}} + \Lambda - \frac{1}{N} \mathbf{1}^T \Lambda \mathbf{1}) \tag{98}$$

Hence, simply by equating factors, we find:

$$wa = 1, \qquad \Lambda = N\lambda \boldsymbol{y} \tag{99}$$

Summing the indices on the second equation tells us:

$$\frac{1}{N} \sum_i [-(\bar{x}^{[i]} - \theta)]_+ = \lambda \tag{100}$$

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:

$$\theta = \frac{\lambda}{q} - |\min \bar{\boldsymbol{x}}| \tag{101}$$

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 Equation (72) we see that a condition is that:

$$|\min \bar{\boldsymbol{x}} - wa(\theta - \lambda)| = \lambda \frac{1-q}{q} \leq \lambda \tag{102}$$

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.

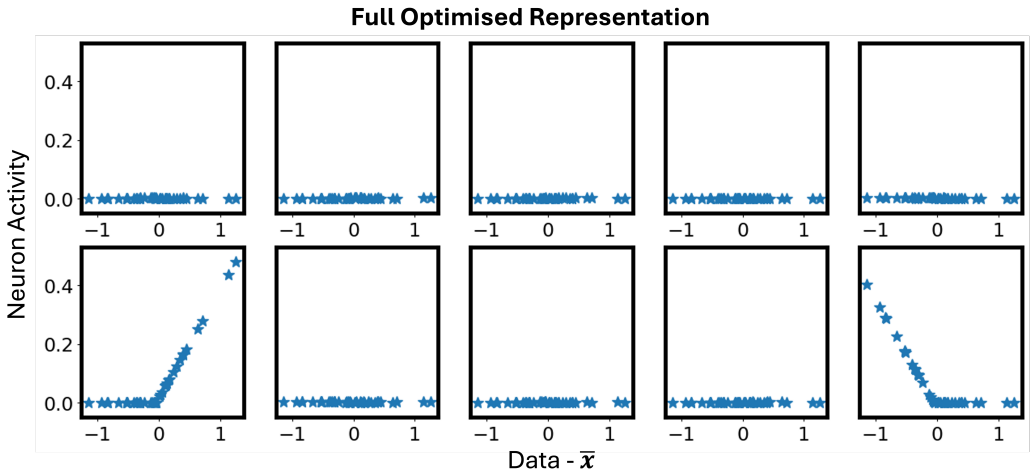

Figure 5. We optimise ten nonnegative neurons to encode a scalar variable by minimising Equation (4). Only two neurons have non-zero activity and their functional form matches the prediction, Figure 3. Somehow optimisation selects this solution from amongst the degenerate set of optimal solutions.

E.5. Empirical Verification

We empirically test these claims by optimising a nonnegative encoding to optimise Equation (4). We input a scalar and run gradient descent on $\boldsymbol{Z}$ subject to nonnegativity. Figure 3 shows the comparison of the two non-zero neurons to the predictions from Section E.3. We include the full 10 neuron representation we optimised to illustrate how cleanly the representation uses only two neurons, Figure 5. The choice of using two neurons, rather than another of the degenerately optimal solutions, somehow results from optimisation.

F. More Features than Data 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, Section A.4, and in for a related problem one-neuron-per-datapoint is a global minima (Bach et al., 2008). However, this problem has a slight non-triviality: we saw in Section 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:

$$z_d^{[i]} = \begin{cases} a_d \geq 0 & \text{if} \quad d = i \\ 0 & \text{else} \end{cases} \tag{103}$$

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

$$\sum_d ||\bar{\boldsymbol{x}}_d - \boldsymbol{b}||^2 \left(1 - \frac{a_d}{\bar{\boldsymbol{x}}_d - \boldsymbol{b}}\right)^2 + 2\lambda \sum_d a_d \tag{104}$$

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 $\bar{\boldsymbol{x}}_d - \boldsymbol{b}$ for the $\bar{\boldsymbol{x}}_d$ on which it is active. This constrains the active sets of each neuron to those along rays from $\boldsymbol{b}$ and tells us that the global minima is exactly the set of ray solutions.

Optimising with respect to a_d and taking into account the nonnegativity we get:

$$a_d = \max(||\bar{\boldsymbol{x}}_d - \boldsymbol{b}|| - \lambda, 0) \tag{105}$$

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:

$$\boldsymbol{b} = \arg\min_b \sum_d \ell_\lambda(||\bar{\boldsymbol{x}}_d - \boldsymbol{b}||) \quad \text{where} \quad \ell_\lambda(||\bar{\boldsymbol{x}}_d - \boldsymbol{b}||) = \begin{cases} ||\bar{\boldsymbol{x}}_d - \boldsymbol{b}||^2 & ||\bar{\boldsymbol{x}}_d - \boldsymbol{b}|| \leq \lambda \\ 2\lambda||\bar{\boldsymbol{x}}_d - \boldsymbol{b}|| - \lambda^2 & ||\bar{\boldsymbol{x}}_d - \boldsymbol{b}|| > \lambda \end{cases} \tag{106}$$

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

1D Dataset We derived the representation of 1D data using a convoluted KKT argument, Section 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.